CONTINUOUS FINITE ELEMENT APPROXIMATION OF HYPERBOLIC SYSTEMS

A Dissertation

by

YONG YANG

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Jean-Luc Guermond
Committee Members,	Bojan Popov
	Andrea Bonito
	Guergana Petrova
	Jean Concetto Ragusa
Head of Department,	Emil Straube

August 2016

Major Subject: Mathematics

ABSTRACT

The main purpose of this work is to study continuous finite element methods for hyperbolic problems. In scalar case, it is shown that using consistent mass matrix is not compatible with the maximum principle. Moreover, we propose two algorithms which preserve the maximum principle and have high order convergence at the same time. For hyperbolic systems, such as Euler equations, we propose two methods which keep the invariant domain property even in Arbitrary Lagrangian Eulerian (ALE) framework.

To my family

ACKNOWLEDGEMENTS

First of all, I would like to thank my advisors Professor Jean-Luc Guermond and Professor Bojan Popov for their guidance on this research work. Thanks for their stimulating and valuable patience and encouragement. I would have not finished this work without their continued advice, assistance and support.

I would like to thank my friends Laura Saavedra, Murtazo Nazarov, Daniel Castanonquiroz, and Manuel Quezada. Thanks for their suggestions and helps.

Special thanks to many other faculty members in the department. They provided outstanding courses and helped me a lot to understand deep theories. Their persistence on the research always encouraged me to work hard and never give up.

I would also like to extend my gratitude to my parents for their understanding and unconditional support throughout my study.

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

Many physical applications are modeled by hyperbolic systems of conservation laws. For example, the Euler equations are used to describe flows of inviscid compressible fluids (see, e.g., [74], [69, p. 346]); the Buckley-Leverett equation is used to model a two-phase flow in porous media (see, e.g., [50, p. 239]); Magnetohydrodynamics (MHD) is used to model electrically conducting fluids such as plasmas and liquid metals (see, e.g., [55, Chapter 5], [27]). All of these are examples of nonlinear hyperbolic systems of conservation laws.

The difficult and interesting part when solving such nonlinear problems lies in the fact that discontinuities may appear in finite time in the solutions even if the initial conditions are smooth.

The continuous finite element method (FEM) (see, e.g., [29, 73, 7, 21]) has been used widely in the literature as a numerical method to approximate a variety of problems including hyperbolic conservation laws (see, e.g., [71, 9, 10, 8, 40, 4]).

1.1 Motivation and outline

In this dissertation we want to address and answer three questions on using continuous finite element methods to solve hyperbolic problems:

1. Can we use the consistent mass matrix in a finite element method and preserve the maximum principle of the partial differential equation?

2. For a scalar conservation law, can we achieve high order accuracy and keep the maximum principle at the same time?

3. For a nonlinear hyperbolic system, can we construct a conservative FEM in Arbitrary Lagrangian Eulerian (ALE) framework and preserve the invariant domain property of the system at the same time?

These three questions are fundamental in the design of continuous finite element methods for hyperbolic problems and they are worth studying in details.

For scalar conservation laws, the entropy solution satisfies a maximum principle. It is common to impose a local maximum principle to a numerical method to avoid unphysical under- and over-shoots. It is well-known that Galerkin approximation, even for steady transport diffusion problems, may produce large oscillations. This is because of the negative dissipation introduced by the Galerkin formulation (see, e.g., [20, Remark 2.6]). It is also the reason why additional stabilization have to be included in FEM when solving these kinds of problems. Those techniques include adding artificial numerical diffusion or using upwind approximation of the convective term. Some well-known methods include Streamline-Upwind Petrov-Galerkin method , Galerkin/least-squares method, characteristic-Galerkin method and Taylor-Galerkin method (see, e.g., [20, 82], [54, §2.6], [24, p. 346], and the references therein). However, they are designed for steady problems and most of the time are directly applied to unsteady problems without justification of the maximum principle even for scalar conservation laws (see, e.g., [21, 20, 82]). A common technique to preserve the maximum principle is to use the lumped mass matrix, instead of the consistent mass matrix. Mathematically, it allows for a simple proof of the maximum principle (see, e.g., [45] or Lemma 2.1.5, Lemma 2.2.7). However, at least for piecewise linear approximation, it is well-known that lumping the mass matrix induces dispersion errors that have adverse effects when solving transport-like equations with non-smooth initial data (see, e.g., [45]). A natural question is whether it is possible to keep the maximum principle when the consistent mass matrix is used. In Section 2, we will show that a continuous finite element method based on artificial viscosity in space and explicit time stepping cannot satisfy the maximum principle for 1D unsteady transport equations if the consistent mass matrix is used, see Theorem 2.1.4 and Theorem 2.2.6. In fact, the same conclusion holds for any 1D nonlinear conservation laws, see [45, Theorem 3.2 and Theorem 4.3].

The second question addressed in this dissertation is on preserving the maximum principle and high-order accuracy in space at the same time for scalar conservation laws. As stated in [61], "... one unavoidable difficulty in numerical computations of discontinuous

solutions of conservation laws is that either the method is first-order accurate on smooth flows and the discontinuities are excessively smeared, or else spurious oscillations are introduced which pollute the solution and sometimes lead to nonlinear instability...", those two properties are difficult to achieve at the same time. For instance, it is known that for 1D scalar problems Godunov's method satisfies the maximum principle but is only first order accurate in space. On the other hand, the Lax-Wendroff method is second order accurate but it does not satisfy the maximum principle, and there are spurious oscillations present. One explanation of this phenomenon is that the phase velocity is smaller and all the waves for different frequency travel at different speed, leading to dispersion and an oscillation wave lagging behind the discontinuity (see, e.g., [60, (11.15)]). Hence it is interesting to address these two questions and construct a method which keeps the maximum principle and has high order convergence at the same time. In Section 3, we propose a algorithm which is maximum principle preserving, see Theorem 3.3.6 and Theorem 3.6.6, and is convergent to the unique entropy solution with high-order accuracy for any scalar conservation laws on any unstructured meshes, see (3.51). The four key ingredients of this new method are the first-order technique introduced in [36], a novel treatment of the consistent mass matrix from [39], a high-order approximation (entropy-viscosity method of [40]) and the Boris-Book-Zalesak flux correction technique (see, e.g., [5, 80]). The main characteristics of the new method are: (i) it is maximum principle preserving, (ii) it preserves the second order accuracy, (iii) the dispersion errors induced by the mass lumping step are corrected in the flux limiting step. A generalized Zalesak limiter is presented in (3.57) which has the same properties as the original limiter, see Theorem 3.7.3.

The third question addressed in Section 4 is about nonlinear hyperbolic systems. The analogue of the maximum principle for hyperbolic systems is the so-called invariant domain property (see, e.g., [14, 48, 49, 25, 6, 43, 44]). In fact, for a scalar conservation law the maximum principle is equivalent to claim that a closed interval is a invariant domain. For instance, the set $\{\mathbf{u} := (\rho, \mathbf{m}, E)^T : \rho \geq 0, E - \frac{\mathbf{m}^2}{2\rho} \geq 0, s \geq r\}$ for any $r \in \mathbb{R}$ is convex with respect to the conservative variables $\mathbf{u}$ and is an invariant domain of the Euler

equations, where ρ is the density, $\mathbf{m}$ is the momentum, E is the total energy, and s is the special entropy of the system, see [43, (2.15)]. Furthermore, we are going to work in the ALE framework, because as it is stated in [64], the ALE methodology combines the best features of Lagrangian and Eulerian representations in order to obtain a flexible and robust solution algorithm since purely Lagrangian methods tend to tangle and distort the mesh and purely Eulerian methods are more diffusive in contact regions. There are two types of ALE frameworks: (i) Lagrangian + Rezoning + Remapping (see, e.g., [64]); (ii) unsplit formulation where the mesh motion is built into the system and solved simultaneously (see, e.g., [81, 3, 76, 23]). In Section 4 we will propose two algorithms using the unsplit ALE formulation, see (4.50) and (4.89), which preserves all the convex invariant domains of the underlying system, see Theorem 4.6.9 and Theorem 4.6.20, and can be applied to any hyperbolic system on any unstructured mesh in any space dimension. Both of them also satisfy the following important properties: conservation (see Theorem 4.6.1 and Theorem 4.6.18), discrete entropy inequality (see Theorem 4.6.14 and Theorem 4.6.21) and the so-called Discrete Geometric Conservation Law (DGCL) (see Theorem 4.6.6 and Theorem 4.6.19). Several numerical tests on the Euler equations are presented in §4.7 to confirm those theoretical properties.

2. THE CONSISTENT MASS MATRIX AND THE MAXIMUM PRINCIPLE

In this Section, we will study the necessity of using the lumped matrix in continuous finite element methods to get the maximum principle. We will consider one dimensional (1D) transport equations in two cases. The first one is the Cauchy problem

$$\begin{cases} \partial_t u(x,t) + \beta \partial_x u(x,t) = 0, \quad x \in \Omega = \mathbb{R}, \\ u(x,0) = u^0(x), \end{cases} \tag{2.1}$$

and the second one is the periodic boundary value problem

$$\begin{cases} \partial_t u(x,t) + \beta \partial_x u(x,t) = 0, \quad x \in \Omega = (0,1), \\ u(x,0) = u^0(x), \\ u(0,t) = u(1,t), \end{cases} \tag{2.2}$$

where $\beta \in \mathbb{R}$ is a given constant.

We will show that if one uses continuous $\mathbb{P}_1$ finite element method and the forward Euler method for time stepping to solve the above two problems with adding $-\nabla\cdot(\nu h \nabla u)$ numeral viscosity for stabilization, then the consistent mass matrix cannot be used in order to keep maximum principle, see Theorem 2.1.4 and Theorem 2.2.6, where ν is an arbitrary piecewise constant function. Note that using the lumped mass matrix, defined in (2.19), is feasible to get the maximum principle under certain CFL condition, see Lemma 2.1.5 and Lemma 2.2.7.

As for general nonlinear conservation laws

$$\begin{cases} \partial_t u(x,t) + \partial_x f(u(x,t)) = 0, \quad x \in \Omega \subset \mathbb{R}^1, \\ u(x,0) = u^0(x), \end{cases} \tag{2.3}$$

all conclusions obtained in this Section for the transport equations still holds by generalizing the proofs presented here to the nonlinear conservation laws (see [45] for more details).

2.1 Cauchy problem

For the Cauchy problem (2.1) we assume that there exists $a < b$ and $u_a^0, u_b^0 \in \mathbb{R}$ such that $u^0(x) = u_a^0$ for all $x < a$ and $u^0(x) = u_b^0$ for all $x > b$. Using continuous finite element method to solve a Cauchy problem, we will choose the computation domain Ω_{comp} such that $(a, b) \subset \Omega_{\mathsf{comp}} \subsetneq \mathbb{R}$.

2.1.1 Mesh and finite element space

The mesh $\mathcal{T}_h$ is assumed to be uniformed, i.e., $2N+1$ nodal points, denoted by $\{a_i, i = -N, \cdots, N\}$, are equidistributed over $\overline{\Omega}_{\mathsf{comp}}$ and $\mathcal{T}_h = \{I_i \subset \overline{\Omega}_{\mathsf{comp}} : I_i = [a_i, a_{i+1}], \overline{\Omega}_{\mathsf{comp}} = \bigcup_{i=-N}^{N-1} I_i, |I_i| = \frac{|\Omega_{\mathsf{comp}}|}{2N} = h, i = -N, \cdots, N-1\}$. Each interval I_i has two vertices. In order to label it locally, we introduce a connectivity map $\mathsf{j}^{\mathsf{geo}} : \mathcal{T}_h \times \{1, 2\} \to \{-N, \cdots, +N\}$ which means that the interval I_i has two vertices $a_{\mathsf{j}^{\mathsf{geo}}(i,1)}$ and $a_{\mathsf{j}^{\mathsf{geo}}(i,2)}$ or $a_{\mathsf{j}^{\mathsf{geo}}(I_i,1)}$ and $a_{\mathsf{j}^{\mathsf{geo}}(I_i,2)}$. In fact, for this 1D mesh, since the vertices is ordered, we have that $a_{\mathsf{j}^{\mathsf{geo}}(i,1)} = a_i$ and $a_{\mathsf{j}^{\mathsf{geo}}(i,2)} = a_{i+1}$. The map $\mathsf{j}^{\mathsf{geo}}$ is introduced here for it will be used for any types of finite element in any dimensional space in the rest of this dissertation (see, e.g. Section 4.3). Define $S_i := \mathrm{supp}(\phi_i)$. In 1D case, we have $S_i = I_{i-1} \cup I_i$.

Introduce the trial space V_h as the set of continuous piecewise linear functions

$$V_h = \{v_h \in \mathcal{C}^0(\overline{\Omega}_{\mathsf{comp}}) : v_h|_{I_i} \in \mathbb{P}_1, \forall i = -N, \cdots, N-1\}. \tag{2.4}$$

It is a vector space with dimension $2N+1$. Let $\{\varphi_{-N}, \ldots, \varphi_N\}$ be the Lagrange nodal basis associated with the Lagrange nodes of the mesh $\mathcal{T}_h$, i.e., $\varphi_i(a_j) = \delta_{ij}$. We also introduce a space for the artificial viscosity

$$D_h = \{v_h \in L^\infty(\Omega_{\mathsf{comp}}); v_h \circ|_{I_i} \in \mathbb{P}_0, \forall i = -N, \cdots, N-1\}. \tag{2.5}$$

2.1.2 Algorithm

Using the forward Euler method for time stepping, the continuous finite element method used to solve (2.1) is to find $u_h^{n+1} \in V_h$ such that

$$(\frac{u_h^{n+1} - u_h^n}{\Delta t}, v_h) + \beta(\partial_x u_h^n, v_h) + (\nu h \partial_x u_h^n, \partial_x v_h) = 0, \quad \forall v_h \in V_h, \tag{2.6}$$

where $(\cdot, \cdot)$ is the $L^2(\Omega)$ inner product, $\nu(> 0)$ is a given constant, $u_h^n \in V_h$ and V_h is defined in (2.4). Here we have added the artificial viscosity $-\nabla\cdot(\nu h \nabla u)$ to stabilize the method in the same spirit of the method of vanishing viscosity [16, §6.3].

Since the mesh is uniform, in discrete form, the problem (2.6) is equivalent to the following linear system

$$\begin{aligned}\frac{U_{h,i+1}^{n+1} + 4U_{h,i}^{n+1} + U_{h,i-1}^{n+1}}{6} =& \frac{U_{h,i+1}^n + 4U_{h,i}^n + U_{h,i-1}^n}{6} \\ &+ \lambda(-\frac{\beta}{2} + c)U_{h,i+1}^n + \lambda(-2c)U_{h,i}^n + \lambda(\frac{\beta}{2} + c)U_{h,i-1}^n,\end{aligned} \tag{2.7}$$

where $u_h^n(x) := \sum_{i=-N}^{N} U_{h,i}^n \varphi_i(x)$ and $\lambda := \frac{\Delta t}{h}$.

To get u_h^0 from u^0, depending on the regularity of u^0, we can choose its L^2 projection $P_h u_h^0$ or its Lagrange interpolation $I_h u_h^0$, where $P_h : L^2 \to V_h$ is defined by $(P_h u, \phi) = (u, \phi)$ for all $\phi \in V_h$ and $I_h : C \to V_h$ is defined by $I_h u = \sum u(a_i)\phi_i$.

2.1.3 Main result

For $u_h^b \in V_h$, we introduce the notation $U_{h,i}^n$ which satisfies

$$u_h^n = \sum_{i=-N}^{N} U_{h,i}^n \varphi_i(x). \tag{2.8}$$

Choosing a special initial data

$$u_0(x) = 1 - H(x), \tag{2.9}$$

where $H(\cdot) = \mathbb{1}_{x>0}$ is the Heaviside function, and using its Lagrange interpolation to get u_h^0 in (2.6) as follows

$$u_h^0(x) = I_h u_0(x) := \sum_{i=-N}^{N} u_0(x_i)\varphi_i(x). \tag{2.10}$$

we obtain the following main Theorem when ν is a constant.

Theorem 2.1.1. *For any given constant ν and $\Delta t^1 > 0$, the solution u_h^1 of the scheme* (2.6) *with initial data* (2.10) *will violates the maximum principle, i.e.,*

$$\max_i\{U_{h,i}^1\} > \max_i\{U_{h,i}^0\} \quad \textit{and} \quad \min_i\{U_{h,i}^1\} < \min_i\{U_{h,i}^0\}. \tag{2.11}$$

Proof. Using the notation in (2.8), we have

$$U_{h,i}^0 = \begin{cases} 1 & \text{for} \quad i \le 0 \\ 0 & \text{for} \quad i > 0. \end{cases}$$

From (2.6), it follows that u_h^1 satisfies the following equation

$$\frac{U_{h,i+1}^1 + 4U_{h,i}^1 + U_{h,i-1}^1}{6} = \begin{cases} 1 & \text{for} \quad i < 0 \\ \frac{5}{6} + \lambda(\frac{\beta}{2} - c) & \text{for} \quad i = 0 \\ \frac{1}{6} + \lambda(\frac{\beta}{2} + c) & \text{for} \quad i = 1 \\ 0 & \text{for} \quad i > 1, \end{cases} \tag{2.12}$$

where $\lambda := \frac{\Delta t^1}{h}$. First, let us solve the above equations for $i > 1$,

$$\frac{U_{h,i+1}^1 + 4U_{h,i}^1 + U_{h,i-1}^1}{6} = 0.$$

Suppose $U_{h,i}^1 = \alpha r^i$ for $i \ge 1$. It follows that

$$r^2 + 4r + 1 = 0.$$

Then we get two solutions for r:

$$r_+ = -2 + \sqrt{3}, \quad r_- = -2 - \sqrt{3}.$$

Therefore,

$$U^1_{h,i} = \alpha_1 r_+^i + \alpha_2 r_-^i,$$

for some constants α_1 and α_2.

Since $|r_-| > 1$ and $|r_-^i| \to +\infty$ and $U^1_{h,i} \to 0$ as $i \to N$(N is taken to be large enough), it follows that

$$\alpha_2 = 0,$$

and thus we obtain that

$$U^1_{h,i} = \alpha_1 r_+^i, \quad i \geq 1. \tag{2.13}$$

Then, solving the following equations in the same spirit

$$\frac{U^1_{h,i+1} + 4U^1_{h,i} + U^1_{h,i-1}}{6} = 1, \quad \forall i < 0,$$

we have (since $|r_+^i| \to +\infty$ and $U^1_{h,i} \to 1$ as $i \to -N$(N is taken to be large enough))

$$U^1_{h,i} = \beta_2 r_-^i + 1, \quad i \leq 0. \tag{2.14}$$

For α_1 and β_2, there are two additional equations for us to use

$$\begin{cases} \frac{U^1_{h,1}+4U^1_{h,0}+U^1_{h,-1}}{6} = \frac{5}{6} + \lambda(\frac{\beta}{2} - c) \\ \frac{U^1_{h,2}+4U^1_{h,1}+U^1_{h,0}}{6} = \frac{1}{6} + \lambda(\frac{\beta}{2} + c). \end{cases} \tag{2.15}$$

Plugging (2.13), (2.14) into (2.15), we have

$$\begin{cases} \alpha_1 r_+ + 4(\beta_2 + 1) + (\beta_2 r_-^{-1} + 1) = 5 + 6\lambda(\frac{\beta}{2} - c) \\ \alpha_1 r_+^2 + 4\alpha_1 r_+ + (\beta_2 + 1) = 1 + 6\lambda(\frac{\beta}{2} + c), \end{cases}$$

i.e.,

$$\begin{cases} \alpha_1(-2+\sqrt{3}) + \beta_2(2+\sqrt{3}) = 6\lambda\left(\frac{\beta}{2} - c\right) \\ -\alpha_1 + \beta_2 = 6\lambda\left(\frac{\beta}{2} + c\right). \end{cases}$$

Solving these equations, we obtain the solution u_h^1 as follows

$$U_{h,i}^1 = \begin{cases} \sqrt{3}\lambda[(-1-\sqrt{3})\beta/2 + (-3-\sqrt{3})c]r_+^i & \text{for} \quad i \geq 1 \\ \sqrt{3}\lambda[(-1+\sqrt{3})\beta/2 + (-3+\sqrt{3})c]r_-^i + 1 & \text{for} \quad i \leq 0. \end{cases} \tag{2.16}$$

Since $\max_i\{U_{h,i}^0\} = 1, \min_i\{U_{h,i}^0\} = 0$, and both $r_\pm < 0$ in (2.16), one can see that u_h^1 shows undershoots and overshoots at the same time. Therefore it violates the maximum principle and (2.11) holds. □

Remark 2.1.2. *If ν is constant, by applying Fourier transformation to both sides of $\partial_t u + \beta\partial_x u = \nu h\partial_{xx} u$, we see that the constant ν must be positive to make the method stable since the Fourier transform $\hat{u}(t,\omega)$ satisfies*

$$\hat{u}(t,\omega) = \exp(-i\omega\beta t - \nu h\omega^2 t)\hat{u}^0(\omega)$$

for any $\omega \in \mathbb{R}$.

The same conclusion holds for any $\nu \in D_h$.

Theorem 2.1.3. *For any $\nu \in D_h$ and $\Delta t^1 > 0$, the solution u_h^1 to the problem* (2.6) *with initial data* (2.10) *will violate the maximum principle, i.e.,*

$$\max_i\{U_{h,i}^1\} > \max_i\{U_{h,i}^0\},$$

and

$$\min_i\{U_{h,i}^1\} < \min_i\{U_{h,i}^0\}.$$

Proof. Denote $\nu|_{I_i} = \nu_{i+\frac{1}{2}}$. We can use a similar proof as Theorem 2.1.1. The only

difference is to replace the equation(2.12) by

$$\frac{U^1_{h,i+1}+4U^1_{h,i}+U^1_{h,i-1}}{6}=\begin{cases} 1 & \text{for} \quad i<0 \\ \frac{5}{6}+\lambda\left(\frac{\beta}{2}-\nu_{i+\frac{1}{2}}\right) & \text{for} \quad i=0 \\ \frac{1}{6}+\lambda\left(\frac{\beta}{2}+\nu_{i+\frac{1}{2}}\right) & \text{for} \quad i=1 \\ 0 & \text{for} \quad i>1, \end{cases} \tag{2.17}$$

□

Theorem 2.1.4. *For any $\nu \in D_h$ and $\Delta t^1 > 0$, there exists $u^0 \in V_h$ (V_h is defined in* (2.4)*) such that the solution u^1 violates the maximum principle at t^1, i.e.,*

$$\max_i\{u_i^1\} > \max_i\{u_i^0\} \quad \textit{and} \quad \min_i\{u_i^1\} < \min_i\{u_i^0\}.$$

2.1.4 Using lumped mass matrix

What is the mass lumping? Mathematically, it means the use of a quadrature to approximate the integral $[M_C]_{ij} := \int \phi_i\phi_j$ (see, e.g., [73, p. 240]). Take $\mathbb{P}_1$ finite element on a mesh $\mathcal{T}_h$ with triangles in $\mathbb{R}^2$ as an example. The lumped mass matrix M_L is defined by

$$[M_L]_{ij} := (\phi_i,\phi_j)_L := \sum_{\mathcal{T}_h \ni K \subset \mathcal{S}_i \cap S_j} \frac{|K|}{3} \sum_{i=1,2,3} \phi_i(\mathbf{a}_{\mathrm{j}^{\mathrm{geo}}(K,i)})\phi_j(\mathbf{a}_{\mathrm{j}^{\mathrm{geo}}(K,i)}), \tag{2.18}$$

where ϕ_i is the global shape function corresponding to the node $\mathbf{a}_i$, $\{\mathbf{a}_i\}$ is the collection of all Lagrangian nodes in the mesh $\mathcal{T}_h$ and $S_i := \mathrm{supp}(\phi_i)$.

By (2.18), we have that

$$[M_L]_{ij} = \delta_{ij}\int \phi_i = \delta_{ij}\sum_j \int \phi_i\phi_j = \delta_{ij}\sum_j [M_C]_{ij},$$

which means that mass lumping is equivalent to the so-called "row-sum" technique.

If the problem is posed in one dimensional space, then one can introduce the lumped

mass matrix as follows

$$[M_L]_{ij} = (\phi_i, \phi_j)_L := \sum_{\mathcal{T}_h \ni K \subset \mathcal{S}_i \cap S_j} \frac{|K|}{2} \sum_{i=1,2} \phi_i(\mathbf{a}_{\mathrm{jgeo}(K,i)}) \phi_j(\mathbf{a}_{\mathrm{jgeo}(K,i)}). \tag{2.19}$$

Note that in 1D case, the mass lumping is also related to the Lax-Wendroff finite difference method (see, e.g., [65, (7.8)]).

Numerically, mass lumping is to use a diagonal matrix M_L to approximate the consistent mass matrix M_C which is sparse, banded, and has dense inverse. Note that although renumbering the degree of freedoms helps to reduce the bandwidth of M_C, it is NP-complete. Two widely used algorithms for renumbering includes the reverse Cuthill-McKee algorithm and the Gibbs-Poole-Stockmeyer algorithm (see, e.g. [28]). Since M_L is a diagonal matrix, its inverse can be computed explicitly. This is the reason why M_L is used widely. The "row-sum" technique works for $\mathbb{Q}_1$ finite elements. That is

$$\begin{aligned} [M_L]_{ij} :=& (\phi_i, \phi_j)_L \\ :=& \sum_{K \subset \mathcal{S}_i \cap S_j} \frac{|K|}{4} \sum_{\mathbf{a}_k \in K} \phi_i(\mathbf{a}_k) \phi_j(\mathbf{a}_k) \\ =& \delta_{ij} \int \phi_i. \end{aligned} \tag{2.20}$$

However, it is not always positive for other finite elements since the diagonal terms in M_L is not always positive such as $\mathbb{P}_2$ finite elements (see, e.g., [35, p. 107][39]).

Using the mass lumping to the first term in (2.6), the new algorithm is to find $u_h^{n+1} \in V_h$ such that

$$\left(\frac{u_h^{n+1} - u_h^n}{\Delta t}, v_h \right)_L + \beta(\partial_x u_h^n, v_h) + (\nu h \partial_x u_h^n, \partial_x v_h) = 0, \quad \forall v_h \in V_h. \tag{2.21}$$

The main result of the mass lumping is that it allows the scheme to keep the maximum principle.

Lemma 2.1.5. *If $\nu \in D_h$ and Δt satisfies the condition that*

$$\begin{cases} |\beta| \le 2\min_i \nu_{i+1/2}, \\ \lambda \max_i(\nu_{i-1/2} + \nu_{i+1/2}) \le 1, \end{cases} \tag{2.22}$$

then the solution of (2.21) *satisfies the local maximum principle, i.e.,*

$$\min\{U^n_{h,i-1}, U^n_{h,i}, U^n_{h,i+1}\} \le U^{n+1}_{h,i} \le \max\{U^n_{h,i-1}, U^n_{h,i}, U^n_{h,i+1}\}, \quad \forall n, \forall i. \tag{2.23}$$

Proof. Since V_h is finite dimensional and $V_h = \text{span}\{\phi_i\}$, to solve (2.21) is equivalent to find $u^{n+1}_h \in V_h$ such that

$$\left(\frac{u^{n+1}_h - u^n_h}{\Delta t}, \phi_i\right)_L + \beta(\partial_x u^n_h, \phi_i) + (\nu h \partial_x u^n_h, \partial_x \phi_i) = 0, \quad \forall i. \tag{2.24}$$

That is

$$U^{n+1}_{h,i} = (1 - \nu_{i-1/2}\lambda - \nu_{i+1/2}\lambda)U^n_{h,i} + \lambda\left(-\frac{\beta}{2} + \nu_{i+1/2}\right)U^n_{h,i+1} + \lambda\left(\frac{\beta}{2} + \nu_{i-1/2}\right)U^n_{h,i-1}, \quad \forall i.$$

Since under the condition (2.22), $U^{n+1}_{h,i}$ becomes a convex combination of $U^n_{h,i-1}, U^n_{h,i}$ and $U^n_{h,i+1}$ and hence it satisfies (2.23). □

2.2 Periodic boundary value problem

In this Section, we will use continuous finite element method to solve (2.2) in $\Omega_{\mathsf{comp}} = \Omega = [0, 1]$.

2.2.1 Mesh and finite element space

The mesh $\{\mathcal{T}_h\}$ is assumed to be uniformed, i.e., $N + 1$ nodal points, denoted by $\{a_i := \frac{i}{N}, i = 0, \cdots, N\}$, are equidistributed over $\overline{\Omega} = [0, 1]$ and $\mathcal{T}_h = \{I_i \subset \overline{\Omega} : I_i = [a_i, a_{i+1}], \overline{\Omega}_N = \bigcup_{i=0}^{N-1} I_i, |I_i| = \frac{|\Omega|}{N} = h, i = 0, \cdots, N-1\}$.

Considering the periodic boundary condition in the problem (2.2), we introduce the

trial space V_h as the set of periodic continuous piecewise linear function

$$V_h = \{v_h \in \mathcal{C}^0(\overline{\Omega}) : v_h(0) = v_h(1), v_h|_{I_i} \in \mathbb{P}_1, \forall i = 0, \cdots, N-1\}. \tag{2.25}$$

It is clear that $\dim V_h = N$ and

$$V_h = \text{span}\{\phi_0 + \phi_N, \phi_1, \cdots, \phi_{N-1}\}.$$

Note that ϕ_0 is not in V_h, but $\phi_0 + \phi_N$ is. This is because the periodic boundary condition is enforced in (2.25).

2.2.2 Finite element approximation

Using finite element method to solve (2.2) is to find

$$u_h^{n+1} \in V_h$$

i.e.,

$$u_h^{n+1}(x) = \sum_{i=0}^{N-1} U_{h,i}^{n+1} \varphi_i(x) + U_{h,0}^{n+1} \varphi_N(x), \tag{2.26}$$

such that

$$\left(\frac{u_h^{n+1} - u_h^n}{\Delta t^n}, v_h \right) + \beta(\partial_x u_h^n, v_h) + (\nu h \partial_x u_h^n, \partial_x v_h) = 0, \tag{2.27}$$

for all $v_h \in V_h$, where V_h is defined in (2.25), and $\nu \in D_h$.

If ν is a constant function, it is expressed equivalently in matrix form as

$$\hat{M} \frac{U^{n+1} - U^n}{\Delta t^n} + \hat{A} U^n = 0,$$

where

$$U^{n+1} = (U_{h,0}^{n+1}, U_{h,1}^{n+1}, \cdots, U_{h,N-1}^{n+1})^{\mathsf{T}}, \tag{2.28}$$

$$\hat{M} := \begin{bmatrix} (\phi_0+\phi_N, \phi_0+\phi_N) & (\phi_1, \phi_0+\phi_N) & \dots & (\phi_{N-1}, \phi_0+\phi_N) \\ (\phi_0+\phi_N, \phi_1) & (\phi_1, \phi_1) & \dots & (\phi_{N-1}, \phi_1) \\ \vdots & \vdots & \ddots & \vdots \\ (\phi_0+\phi_N, \phi_{N-1}) & (\phi_0+\phi_N, \phi_{N-1}) & \dots & (\phi_0+\phi_N, \phi_{N-1}) \end{bmatrix}_{N\times N} \tag{2.29}$$

$$= \frac{h}{6} \begin{bmatrix} 4 & 1 & 0 & \dots & 0 & 1 \\ 1 & 4 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 0 & 0 & \dots & 1 & 4 \end{bmatrix}_{N\times N},$$

and

$$\hat{A} := \begin{bmatrix} a(\phi_0+\phi_N, \phi_0+\phi_N) & a(\phi_1, \phi_0+\phi_N) & \dots & a(\phi_{N-1}, \phi_0+\phi_N) \\ a(\phi_0+\phi_N, \phi_1) & a(\phi_1, \phi_1) & \dots & a(\phi_{N-1}, \phi_1) \\ \vdots & \vdots & \ddots & \vdots \\ a(\phi_0+\phi_N, \phi_{N-1}) & a(\phi_0+\phi_N, \phi_{N-1}) & \dots & a(\phi_0+\phi_N, \phi_{N-1}) \end{bmatrix}_{N\times N}$$

$$= \begin{bmatrix} 2\nu & \beta/2-\nu & 0 & \dots & 0 & -\beta/2-\nu \\ -\beta/2-\nu & 2\nu & \beta/2-\nu & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \beta/2-\nu & 0 & 0 & \dots & -\beta/2-\nu & 2\nu \end{bmatrix}_{N\times N},$$

That is

$$MU^{n+1} = AU^n, \tag{2.30}$$

where

$$M = \frac{1}{6} \begin{bmatrix} 4 & 1 & 0 & \dots & 0 & 1 \\ 1 & 4 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 0 & 0 & \dots & 1 & 4 \end{bmatrix}_{N\times N},$$

and

$$
\begin{aligned}
A =& \frac{1}{h}\hat{M} + \frac{\Delta t^n}{h}\hat{A} \qquad (2.31)\\
=& M + \lambda \begin{bmatrix}
-2\nu & -\beta/2+\nu & 0 & \dots & 0 & \beta/2+\nu \\
\beta/2+\nu & -2\nu & -\beta/2+\nu & \dots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
-\beta/2+\nu & 0 & 0 & \dots & \beta/2+\nu & -2\nu
\end{bmatrix}_{N\times N}.
\end{aligned}
$$

Remark 2.2.1. *The expressions in* (2.29) *and* (2.31) *show that M and A are circulant matrices, which are denoted by (see [18])*

$$M = \text{circ}\left[\frac{4}{6}, \frac{1}{6}, 0, \cdots, 0, \frac{1}{6}\right],$$

and

$$A = \text{circ}\left[\frac{4}{6} - 2\nu\lambda, \frac{1}{6} - \frac{\beta}{2}\lambda + \nu\lambda, 0, \cdots, 0, \frac{1}{6} + \frac{\beta}{2}\lambda + \nu\lambda\right]. \qquad (2.32)$$

Remark 2.2.2. *M is invertible since it is a Gram matrix (see, e.g., [17, p. 177]). Furthermore, M^{-1} is a circulant matrix. This is the result of Theorem 3.2.3 of [18] and*

$$
\begin{aligned}
M =& \frac{4}{6}I + \frac{1}{6}\Pi + \frac{1}{6}\Pi^{N-1}\\
=& F^*\left(\frac{4}{6}I + \frac{1}{6}\Omega + \frac{1}{6}\Omega^{N-1}\right)F\\
=& F^*\text{diag}\left(\mathcal{P}_M(1), \mathcal{P}_M(\omega), \cdots, \mathcal{P}_M(\omega^{n-1})\right)F,
\end{aligned}
$$

where $\omega = e^{2\pi i/N}$, $\Omega = \text{diag}\left(1, \omega, \cdots, \omega^{N-1}\right)$,

$$\mathcal{P}_M(z) = \frac{4}{6} + \frac{1}{6}z + \frac{1}{6}z^{N-1},$$

$$\Pi = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 1 & 0 & 0 & \dots & 0 \end{bmatrix}_{N\times N} = F^*\Omega F,$$

and

$$F = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & \bar{\omega} & \dots & \bar{\omega}^{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \bar{\omega}^{N-1} & \dots & \bar{\omega}^{(N-1)(N-1)} \end{bmatrix}_{N\times N}.$$

2.2.3 Main result

In this Section, we want to show the same results as Theorem 2.1.4. It is equivalent to show that $M^{-1}A$ has negative elements for any $\nu \in D_h$ and any $\Delta t^n > 0$.

Lemma 2.2.3. *Assume $N \geq 3$. Then*

$$M^{-1} = \text{circ}[a_0, a_1, \cdots, a_{N-1}], \tag{2.33}$$

where

$$a_j = \sqrt{3}\left(\frac{z_1^j}{1-z_1^N} - \frac{z_2^j}{1-z_2^N}\right), \quad j = 0, 1, \cdots, N-1 \tag{2.34}$$

with

$$z_1 = -2 + \sqrt{3}, \quad z_2 = -2 - \sqrt{3}. \tag{2.35}$$

Proof. Assume $M^{-1} = \text{circ}[a_0, a_1, \cdots, a_{N-1}]$ (see Remark 2.2.2). Since $M^{-1}M = I$, we

get

$$\begin{cases} 4a_0 + a_1 + a_{N-1} = 6, \\ a_{i-1} + 4a_i + a_{i+1} = 0, \quad i = 1, ..., N-1 \\ a_N = a_0. \end{cases} \tag{2.36}$$

The idea is to solve $(N-1)$-equations $a_{i-1} + 4a_i + a_{i+1} = 0, \quad i = 1, ..., N-1$ as in the proof of Theorem 2.1.1, and get

$$a_i = Az_1^i + Bz_2^i,$$

where z_i is defined in (2.35).

Then we use the other two equations to find the right coefficients A and B. In particular, the first equation and the last equation in (2.36) imply that

$$\begin{cases} 4A + 4B + Az_1 + Bz_2 + Az_1^{N-1} + Bz_2^{N-1} = 6, \\ Az_1^N + Bz_2^N = A + B. \end{cases} \tag{2.37}$$

That is

$$\begin{bmatrix} z_1^{N-1} + 4 + z_1 & z_2^{N-1} + 4 + z_2 \\ z_1^N - 1 & z_2^N - 1 \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix} = \begin{bmatrix} 6 \\ 0 \end{bmatrix},$$

or

$$\begin{bmatrix} z_1^{N-1} - z_2 & z_2^{N-1} - z_1 \\ z_1^N - 1 & z_2^N - 1 \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix} = \begin{bmatrix} 6 \\ 0 \end{bmatrix}.$$

Since

$$z_1 z_2 = 1,$$

we obtain that

$$\begin{bmatrix} z_2(z_1^N - 1) & z_1(z_2^N - 1) \\ z_1^N - 1 & z_2^N - 1 \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix} = \begin{bmatrix} 6 \\ 0 \end{bmatrix}.$$

That is

$$\begin{bmatrix} z_2 & z_1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} (z_1^N - 1)A \\ (z_2^N - 1)B \end{bmatrix} = \begin{bmatrix} 6 \\ 0 \end{bmatrix}.$$

Solving it, we get that

$$A = \frac{\sqrt{3}}{1 - z_1^N}, \quad B = \frac{-\sqrt{3}}{1 - z_2^N},$$

which implies (2.34). □

Applying the above lemma, we obtain the following properties of M^{-1}.

Lemma 2.2.4. *$a_j, j = 0, \cdots, N-1$ defined in Lemma 2.2.3 satisfies that*

(i) $a_j = a_k$ *if* $j + k = N$.

(ii) $\operatorname{sgn}(a_j) = \operatorname{sgn}((-1)^j)$ *for* $2j \le N$.

(iii) $|a_j|$ *is decreasing for* $2j \le N$.

Proof. Note that M is symmetric, so is M^{-1}. Since M^{-1} is circulant, the property (i) follows.

By Lemma 2.2.3, if $2j \le N$, then $1 + z_1^{N-2j} > 0$. Since $z_1 < 0$ and $a_j = \frac{1+z_1^{N-2j}}{1-z_1^N} z_1^j$, the property (ii) follows.

Since $|a_j|^2 = \frac{3}{(1-z_1^N)^2}(z_1^j + z_1^{N-j})^2$ and $f(j) := (z_1^j + z_1^{N-j})^2 = (z_1^2)^j + (z_1^2)^{N-j} + 2z_1^N$ is decreasing for $2j \le N$, we have (iii). □

The same results as Theorem 2.1.4 holds when ν is a constant function as stated in the following theorem.

Theorem 2.2.5. *Let $N \ge 6$. For any constant function ν and any $\Delta t^1 > 0$, there exists $u_h^0 \in V_h$ such that the solution u_h^1 of (2.27) violates the maximum principle.*

Proof. In order to make u_h^1 satisfies the maximum principle, i.e., $\max U^1 \le \max U^0$ and $\min U^1 \ge \min U^0$, where U^i is defined in (2.28), $i = 0, 1$, it is necessary to require that

all the elements of $M^{-1}A$ are positive. Indeed, if some element of $M^{-1}A =: (\gamma_{ij})_{N\times N}$ is negative, say $\gamma_{ij} < 0$, then we can take $U^0 = e_j$, which is a unit vector with 1 as j-th element. It follows that $U^1_{h,i} = \sum_{k=0}^{N-1} \gamma_{ik} U^0_k = \gamma_{ij} < 0 = \min_i\{U^0_i\}$.

Therefore, it is sufficient to prove that there exists a negative element in $M^{-1}A$ no matter what ν and λ are.

From (2.32) and (2.33), we obtain that

$$\begin{aligned} M^{-1}A = \mathrm{circ}[&a_{N-1}\left(\frac{1}{6} - \frac{\beta}{2}\lambda + \nu\lambda\right) + a_0\left(\frac{4}{6} - 2\nu\lambda\right) + a_1\left(\frac{1}{6} + \frac{\beta}{2}\lambda + \nu\lambda\right), \\ &a_0\left(\frac{1}{6} - \frac{\beta}{2}\lambda + \nu\lambda\right) + a_1\left(\frac{4}{6} - 2\nu\lambda\right) + a_2\left(\frac{1}{6} + \frac{\beta}{2}\lambda + \nu\lambda\right), \\ &\cdots \\ &a_{N-2}\left(\frac{1}{6} - \frac{\beta}{2}\lambda + \nu\lambda\right) + a_{N-1}\left(\frac{4}{6} - 2\nu\lambda\right) + a_0\left(\frac{1}{6} + \frac{\beta}{2}\lambda + \nu\lambda\right)] \\ =: \mathrm{circ}[&b_0, b_1, \cdots, b_{N-1}]. \end{aligned} \tag{2.38}$$

From (2.35), we know that z_1, z_2 satisfies that $z_i^2 + 4z_i + 1 = 0$ for $i = 1, 2$. By the definition of a_i in (2.34), it follows that

$$\begin{aligned} a_{j-1} + 4a_j + a_{j+1} =& \sqrt{3}\left[\frac{z_1^{j-1}(1 + 4z_1 + z_1^2)}{1 - z_1^N} - \frac{z_2^{j-1}(1 + 4z_2 + z_2^2)}{1 - z_2^N}\right] \\ =& 0, \quad \forall j = 1, \cdots, N-2 \end{aligned} \tag{2.39}$$

which implies that

$$\begin{aligned} b_j =& a_{j-1}\left(\frac{1}{6} - \frac{\beta}{2}\lambda + \nu\lambda\right) + a_j\left(\frac{4}{6} - 2\nu\lambda\right) + a_{j+1}\left(\frac{1}{6} + \frac{\beta}{2}\lambda + \nu\lambda\right) \\ =& \lambda\left[c(a_{j-1} - 2a_j + a_{j+1}) + \frac{\beta}{2}(a_{j+1} - a_{j-1})\right] \\ =& \lambda\left[c(-6a_j) + \frac{\beta}{2}(a_{j+1} - a_{j-1})\right], \end{aligned}$$

for any $j = 1, \cdots, N-2$.

Therefore, if $b_j \geq 0$ for $j = 1, \cdots, N-2$, then ν should satisfy

$$6a_j c \leq \frac{\beta}{2}(a_{j+1} - a_{j-1}), \quad \forall j = 1, \cdots, N-2. \tag{2.40}$$

which is not true. Indeed, let us consider two cases separately.

- Case 1: "$\beta > 0$"

  By Lemma 2.2.4, we can choose k such that $2(k+3) \leq N$ and $a_k > 0$. It follows that $a_k = +\delta_k, a_{k+1} = -\delta_{k+1}, a_{k+2} = +\delta_{k+2}, a_{k+3} = -\delta_{k+3}$, where $\delta_k > \delta_{k+1} > \delta_{k+1} > \delta_{k+3} > 0$. Note that this is possible for $N \geq 6$. From condition (2.40) for $j = k+1$, we obtain that

  $$c \geq \frac{\beta(\delta_k - \delta_{k+2})}{12\delta_{k+1}} > 0. \tag{2.41}$$

  By Lemma 2.2.4, it follows that $a_{N-k-1} = a_{k+1} = -\delta_{k+1}$, $a_{N-k-2} = a_{k+2} = +\delta_{k+2}$, and $a_{N-k-3} = a_{k+3} = -\delta_{k+3}$. Then from condition (2.40) for $j = N-k-2$, we have

  $$c \leq \frac{\beta(-\delta_{k+1} + \delta_{k+3})}{12\delta_{k+2}} < 0, \tag{2.42}$$

  which contradicts (2.41) and (2.42) since there does not exist ν such that both of them are true at the same time.

- Case 2: "$\beta < 0$"

  By Lemma 2.2.4, we can choose k such that $2(k+3) \leq N$ and $a_k < 0$. It follows that $a_k = -\delta_k, a_{k+1} = +\delta_{k+1}, a_{k+2} = -\delta_{k+2}, a_{k+3} = +\delta_{k+3}$, where $\delta_k > \delta_{k+1} > \delta_{k+1} > \delta_{k+3} > 0$. From condition (2.40) for $j = k+1$, we obtain that

  $$c \leq \frac{\beta(\delta_k - \delta_{k+2})}{12\delta_{k+1}} < 0. \tag{2.43}$$

  However, from condition (2.40) for $j = N-k-2$, we obtain that

  $$c \geq \frac{\beta(-\delta_{k+1} + \delta_{k+3})}{12\delta_{k+2}} > 0, \tag{2.44}$$

which contradicts (2.43) and (2.44) since there does not exist ν such that both of them are true at the same time.

Since in both cases there does not exist ν such that all elements of $M^{-1}A$ are positive, we conclude that there exists u_h^0 such that u_h^1 violates the maximum principle. □

For general $\nu \in D_h$ we can obtain the same conclusion by applying a similar proof.

Theorem 2.2.6. *Let $N \geq 6$. For any $\nu \in D_h$ and any $\Delta t^1 > 0$, there exists $u_h^0 \in V_h$ such that the solution u_h^1 violates the maximum principle.*

Proof. Denote $\nu|_{I_i} = \nu_{i+\frac{1}{2}}$. Without loss of generality, let us assume $\beta \geq 0$. As the proof of Theorem 2.2.5, it is sufficient to prove that there exists some negative element in $M^{-1}A$, where $M^{-1} = \mathrm{circ}[a_0, a_1, \cdots, a_{N-1}]$ and

$$A = \begin{bmatrix} \frac{4}{6}-\lambda\nu_{N-\frac{1}{2}}-\lambda\nu_{\frac{1}{2}} & \frac{1}{6}-\lambda\frac{\beta}{2}+\lambda\nu_{\frac{1}{2}} & 0 & \cdots & 0 & \frac{1}{6}+\lambda\frac{\beta}{2}+\lambda\nu_{N-\frac{1}{2}} \\ \frac{1}{6}+\lambda\frac{\beta}{2}+\lambda\nu_{\frac{1}{2}} & \frac{4}{6}-\lambda\nu_{\frac{1}{2}}-\lambda\nu_{\frac{3}{2}} & \frac{1}{6}-\lambda\frac{\beta}{2}+\lambda\nu_{\frac{3}{2}} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{1}{6}-\lambda\frac{\beta}{2}+\lambda\nu_{N-\frac{1}{2}} & 0 & 0 & \cdots & \frac{1}{6}+\lambda\frac{\beta}{2}+\lambda\nu_{N-\frac{3}{2}} & \frac{4}{6}-\lambda\nu_{N-\frac{3}{2}}-\lambda\nu_{N-\frac{1}{2}} \end{bmatrix}_{N\times N}.$$

Since $4a_i + a_{i-1} + a_{i+1} = 6\delta_{i0}$, it follows that

$$\begin{aligned}
(M^{-1}A)_{11} =& 1 + \lambda\nu_{N-\frac{1}{2}}(a_{N-1} - a_0) + \lambda\nu_{\frac{1}{2}}(a_1 - a_0) + \lambda\frac{\beta}{2}(a_1 - a_{N-1}) \\
(M^{-1}A)_{12} =& \quad \lambda\nu_{\frac{1}{2}}(a_0 - a_1) + \lambda\nu_{\frac{3}{2}}(a_2 - a_1) + \lambda\frac{\beta}{2}(a_2 - a_0) \\
(M^{-1}A)_{13} =& \quad \lambda\nu_{\frac{3}{2}}(a_1 - a_2) + \lambda\nu_{\frac{5}{2}}(a_3 - a_2) + \lambda\frac{\beta}{2}(a_3 - a_1) \\
& \cdots \\
(M^{-1}A)_{21} =& \quad \lambda\nu_{N-\frac{1}{2}}(a_0 - a_1) + \lambda\nu_{\frac{1}{2}}(a_2 - a_1) + \lambda\frac{\beta}{2}(a_2 - a_0) \\
(M^{-1}A)_{22} =& 1 + \lambda\nu_{\frac{1}{2}}(a_1 - a_2) + \lambda\nu_{\frac{3}{2}}(a_3 - a_2) + \lambda\frac{\beta}{2}(a_3 - a_1) \\
(M^{-1}A)_{23} =& \quad \lambda\nu_{\frac{3}{2}}(a_2 - a_3) + \lambda\nu_{\frac{5}{2}}(a_4 - a_3) + \lambda\frac{\beta}{2}(a_4 - a_2) \\
& \cdots
\end{aligned}$$

In the analogy of the proof of Theorem 2.2.5, choose k such that $2(k+3) \leq N$ and $a_k > 0$.

It follows that $a_k = +\delta_k, a_{k+1} = -\delta_{k+1}, a_{k+2} = +\delta_{k+2}, a_{k+3} = -\delta_{k+3}$, where $\delta_k > \delta_{k+1} > \delta_{k+1} > \delta_{k+3} > 0$. Consider 4 elements of $M^{-1}A$: $(M^{-1}A)_{k2}, (M^{-1}A)_{(k+1)2}, (M^{-1}A)_{(N-k-3)2}$ and $(M^{-1}A)_{(N-k-2)2}$. The idea is to get some contradiction about $\nu_{\frac{1}{2}}$ and $\nu_{\frac{3}{2}}$ if all such 4 elements are positive. Since

$$\begin{cases} \nu_{\frac{1}{2}}(a_k - a_{k+1}) + \nu_{\frac{3}{2}}(a_{k+2} - a_{k+1}) + \frac{\beta}{2}(a_{k+2} - a_k) & \geq 0 \\ \nu_{\frac{1}{2}}(a_{k+1} - a_{k+2}) + \nu_{\frac{3}{2}}(a_{k+3} - a_{k+2}) + \frac{\beta}{2}(a_{k+3} - a_{k+1}) & \geq 0 \\ \nu_{\frac{1}{2}}(a_{N-k-3} - a_{N-k-2}) + \nu_{\frac{3}{2}}(a_{N-k-1} - a_{N-k-2}) + \frac{\beta}{2}(a_{N-k-1} - a_{N-k-3}) & \geq 0 \\ \nu_{\frac{1}{2}}(a_{N-k-2} - a_{N-k-1}) + \nu_{\frac{3}{2}}(a_{N-k} - a_{N-k-1}) + \frac{\beta}{2}(a_{N-k} - a_{N-k-2}) & \geq 0. \end{cases}$$

by the symmetry of $a_i = a_{N-i}$, we get that

$$\begin{cases} \nu_{\frac{1}{2}}(\delta_k + \delta_{k+1}) + \nu_{\frac{3}{2}}(\delta_{k+2} + \delta_{k+1}) + \frac{\beta}{2}(\delta_{k+2} - \delta_k) & \geq 0 \\ \nu_{\frac{1}{2}}(-\delta_{k+1} - \delta_{k+2}) + \nu_{\frac{3}{2}}(-\delta_{k+3} - \delta_{k+2}) + \frac{\beta}{2}(-\delta_{k+3} + \delta_{k+1}) & \geq 0 \\ \nu_{\frac{1}{2}}(-\delta_{k+3} - \delta_{k+2}) + \nu_{\frac{3}{2}}(-\delta_{k+1} - \delta_{k+2}) + \frac{\beta}{2}(-\delta_{k+1} + \delta_{k+3}) & \geq 0 \\ \nu_{\frac{1}{2}}(\delta_{k+2} + \delta_{k+1}) + \nu_{\frac{3}{2}}(\delta_k + \delta_{k+1}) + \frac{\beta}{2}(\delta_k - \delta_{k+2}) & \geq 0, \end{cases}$$

which is equivalent to

$$\begin{cases} \nu_{\frac{1}{2}}(\delta_k + \delta_{k+1}) + \nu_{\frac{3}{2}}(\delta_{k+2} + \delta_{k+1}) & \geq \frac{\beta}{2}(\delta_k - \delta_{k+2}) \\ \nu_{\frac{1}{2}}(\delta_{k+1} + \delta_{k+2}) + \nu_{\frac{3}{2}}(\delta_{k+3} + \delta_{k+2}) & \leq \frac{\beta}{2}(\delta_{k+1} - \delta_{k+3}) \\ \nu_{\frac{1}{2}}(\delta_{k+3} + \delta_{k+2}) + \nu_{\frac{3}{2}}(\delta_{k+1} + \delta_{k+2}) & \leq -\frac{\beta}{2}(\delta_{k+1} - \delta_{k+3}) \\ \nu_{\frac{1}{2}}(\delta_{k+2} + \delta_{k+1}) + \nu_{\frac{3}{2}}(\delta_k + \delta_{k+1}) & \geq -\frac{\beta}{2}(\delta_k - \delta_{k+2}). \end{cases}$$

Considering the 1st and 4th inequalities, we see that the solution set of $(\nu_{\frac{1}{2}}, \nu_{\frac{3}{2}})$ lies in a cone K_u with vertex at $(\frac{\beta}{2}, -\frac{\beta}{2})$ in Figure 2.1. Similarly for the 2nd and 3rd inequalities, the solution set of $(\nu_{\frac{1}{2}}, \nu_{\frac{3}{2}})$ lies in a cone K_d at the same vertex. Since the only intersection

is that vertex, we obtain that

$$\nu_{\frac{1}{2}} = \frac{\beta}{2}, \quad \nu_{\frac{3}{2}} = -\frac{\beta}{2}.$$

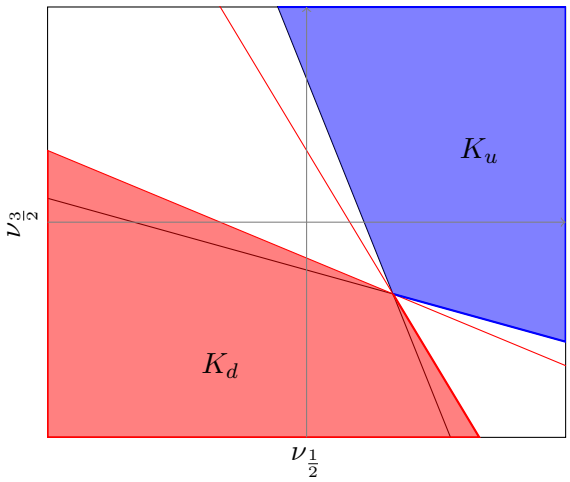


Figure 2.1: Two cones K_u and K_d

Similarly, if choosing other 4 elements in 3rd column of $M^{-1}A$, we obtain that

$$\nu_{\frac{3}{2}} = \frac{\beta}{2}, \quad \nu_{\frac{5}{2}} = -\frac{\beta}{2}.$$

Since $\beta \neq 0$, a contradiction on $\nu_{\frac{3}{2}}$ is obtained, which completes the proof. □

2.2.4 Using lumped mass matrix

Using mass lumping to solve (2.2) is to find $u_h^{n+1} \in V_h$ i.e.,

$$u_h^{n+1}(x) = \sum_{i=0}^{N-1} U_{h,i}^{n+1} \varphi_i(x) + U_{h,0}^{n+1} \varphi_N(x), \tag{2.45}$$

such that

$$\left(\frac{u_h^{n+1}-u_h^n}{\Delta t}, v_h\right)_L + \beta(\partial_x u_h^n, v_h) + (\nu h \partial_x u_h^n, \partial_x v_h) = 0, \quad \forall v_h \in V_h, \tag{2.46}$$

where V_h is defined in (2.25), $(\cdot,\cdot)_L$ is defined in (2.18), and $\nu \in D_h$.

The discrete form is as follows

$$\begin{cases} U_{h,0}^{n+1} = & [1-\lambda(\nu_{N-\frac{1}{2}}+\nu_{\frac{1}{2}})]U_{h,0}^n + \lambda(-\frac{\beta}{2}+\nu_{\frac{1}{2}})U_{h,2}^n + \lambda(\frac{\beta}{2}+\nu_{N-\frac{1}{2}})U_{h,N-1}^n, \\ U_{h,i}^{n+1} = & [1-\lambda(\nu_{i-\frac{1}{2}}+\nu_{i+\frac{1}{2}})]U_{h,i}^n + \lambda(-\frac{\beta}{2}+\nu_{i+\frac{1}{2}})U_{h,i+1}^n + \lambda(\frac{\beta}{2}+\nu_{i-\frac{1}{2}})U_{h,i-1}^n, \\ U_{h,N-1}^{n+1} = & [1-\lambda(\nu_{N-\frac{3}{2}}+\nu_{N+\frac{1}{2}})]U_{h,N-1}^n + \lambda(-\frac{\beta}{2}+\nu_{N+\frac{1}{2}})U_{h,0}^n + \lambda(\frac{\beta}{2}+\nu_{N-\frac{3}{2}})U_{h,N-2}^n, \end{cases} \tag{2.47}$$

where $\lambda := \frac{\Delta t}{h}$.

Lemma 2.2.7. *If $\nu \in D_h$ and $\Delta t^n = \lambda h$ satisfies that*

$$\begin{cases} |\beta| \le 2\min_i \nu_{i+\frac{1}{2}}, \\ \lambda \max_i \left(\nu_{i-\frac{1}{2}}+\nu_{i+\frac{1}{2}}\right) \le 1, \end{cases} \quad , \quad i = 1, \cdots, N-1 \tag{2.48}$$

where $\nu|_{I_i} = \nu_{i+\frac{1}{2}}$ with the convention $\nu_{-\frac{1}{2}} := \nu_{N-\frac{1}{2}}$, then the solution of (2.46) *satisfies the local maximum principle* (2.23).

Proof. From (2.47), it is readily seen that under the condition (2.48), $U_{h,i}^{n+1}$ is a convex combination of $U_{h,i-1}^n, U_{h,i}^n$,and $U_{h,i+1}^n$, which implies the local maximum principle

$$\min\{U_{h,i-1}^n, U_{h,i}^n, U_{h,i+1}^n\} \le U_{h,i}^{n+1} \le \max\{U_{h,i-1}^n, U_{h,i}^n, U_{h,i+1}^n\}, \quad \forall n, \forall i. \tag{2.49}$$

□

3. ZALESAK LIMITER FOR SCALAR CONSERVATION LAWS *

In this Section, we will investigate two continuous $\mathbb{P}_1$ finite element methods for solving scalar conservation laws, see (3.51) and (3.57). Both of them have two good properties: maximum principle preserving and high-order accuracy.

3.1 Maximum principle

The scalar conservation law is usually written as

$$\begin{cases} \partial_t u(\mathbf{x},t) + \nabla \cdot \mathbf{f}(u(\mathbf{x},t)) = 0, \quad (\mathbf{x},t) \in \Omega \times [0,T] \subset \mathbb{R}^d \\ u(\mathbf{x},0) = u_0(\mathbf{x}), \end{cases} \tag{3.1}$$

along with appropriate boundary condition. To simplify the problem without considering boundary issues, we assume that the problem is a Cauchy problem or has periodic boundary condition or the initial data u_0 has a compact support and the final time T is smaller enough such that the influence region of $\text{supp}(u_0)$ does not reach the boundary $\partial\Omega$ for all $t \in [0,T]$.

The maximum principle is an important property of the entropy solution of the scalar hyperbolic conservation law. The entropy solution is introduced, due to the non-uniqueness of weak solutions, to satisfy the so-called entropy inequality

$$\int_0^T \int_{\mathbb{R}^d} [\eta(u)\partial_t\psi + \nabla\psi \cdot \mathbf{q}(u)] \,\mathrm{d}\mathbf{x}\,\mathrm{d}t + \int_\Omega u(\mathbf{x},0)\eta(u(\mathbf{x},0)) \le 0, \tag{3.2}$$

for every convex function η, where $\mathbf{q}(u) = \int^u \eta'(v)\mathbf{f}'(v)\,\mathrm{d}v$, where the test function ψ is positive with compact support and is Lipschitz continuous on $\mathbb{R}^d \times [0,T)$. The maximum principle says that the entropy solution u of (3.1) has the property that

$$u(\mathbf{x},t) \in [m,M] \tag{3.3}$$

*Part of this Section are reprinted, with modification, from [37], "A second-order maximum principle preserving Lagrange finite element technique for nonlinear scalar conservation equations" by Jean-Luc Guermond, Murtazo Nazarov, Bojan Popov and Yong Yang, 2014. *SIAM Journal on Numerical Analysis*, 52(4), 2163–2182. Copyright 2014 by Society for Industrial and Applied Mathematics.

where

$$m := \min_{\mathbf{x}} u(\mathbf{x}, 0), \quad M := \max_{\mathbf{x}} u(\mathbf{x}, 0). \tag{3.4}$$

In particular, this is implied by the property (see, e.g., Theorem 6.2.3 [16]) that for two entropy solutions u and $\bar{u}$ corresponding to two initial conditions u_0 and $\bar{u}_0$, there exist constants R and s such that the following inequality hold

$$\int_{|\mathbf{x}|\le R} [u(\mathbf{x}, t) - \bar{u}(\mathbf{x}, t)]^+ \, \mathrm{d}\mathbf{x} \le \int_{|\mathbf{x}|\le R+st} [u_0(\mathbf{x}) - \bar{u}_0(\mathbf{x})]^+ \, \mathrm{d}\mathbf{x} \tag{3.5}$$

which is proved by using the entropy pairs of Kruzkov (see, e.g., [16, (6.2.6)])

$$\eta(u; \bar{u}) = (u - \bar{u})^+, \quad \mathbf{Q}(u; \bar{u}) = \mathrm{sgn}(u - \bar{u})^+ [\mathbf{f}(u) - \mathbf{f}(\bar{u})], \quad \bar{u} \in \mathbb{R}. \tag{3.6}$$

3.2 Mesh and finite element space

Let $\{\mathcal{T}_h\}$ be a family of conforming (no hanging nodes) and shape regular meshes. Define

$$\underline{h} := \min_{K \in \mathcal{T}_h} h_K,$$

and

$$h_K := \frac{1}{\max_j \sup_{\mathbf{x} \in K} \|\nabla \phi_j(\mathbf{x})\|_{l^2}}.$$

Let $\{\phi_1, \cdots, \phi_N\}$ be the nodal Lagrange basis of V_h associated with the vertices $\{\mathbf{a}_1, \cdots, \mathbf{a}_N\}$ of the mesh $\mathcal{T}_h$, i.e., $\phi_i(\mathbf{a}_j) = \delta_{ij}$. The approximation space or test function space V_h is defined by

$$V_h := \{v \in C^0(\Omega; \mathbb{R}) : v|_K \circ \Phi_K \in \hat{P}, \forall K \in \mathcal{T}_h\}, \tag{3.7}$$

where $\{\hat{K}, \hat{P}, \hat{\Sigma}\}$ is the reference Lagrange finite elements, and $\Phi_K : \hat{K} \to K$ is given for each cell K. If K is a triangle, then Φ_K can be chosen as affine map. However, if K is arbitrary quadrilateral, Φ_K should be a bilinear map, since if Φ_K is an affine map, then K must be a parallelogram (see, e.g., [21, p. 34]).

Define $S_i = \mathrm{supp}(\phi_i)$ and $S_{ij} = \mathrm{supp}(\phi_i) \cap \mathrm{supp}(\phi_j)$. We have $S_i = \cup_{K \subset S_i} K$ and

$S_{ij} = \cup_{K \subset S_{ij}} K$. Introduce $\mathcal{I}(E) = \{i : |S_i \cap \bar{E}| \neq 0\}$ for any subset $E \subset \bar{\Omega}$. Here the cell K is assumed to be closed.

3.3 Low order scheme

In [36], the authors propose a continuous finite element method which satisfies the maximum principle. Using similar argument, we get the following results.

Definition 3.3.1. *Define the bilinear form $b_K(\cdot,\cdot)$ corresponding to each cell $K \in \mathcal{T}_h$ satisfies the following four properties*

(i) $b_K(\phi_j, \phi_i) = 0$, *if* $i \notin \mathcal{I}(K)$ *or* $j \notin \mathcal{I}(K)$,

(ii) $b_K(\phi_j, \phi_i) \sim -|K|$ *if* $i \neq j$, $i, j \in \mathcal{I}(K)$,

(iii) symmetry $b_K(\phi_j, \phi_i) = b_K(\phi_i, \phi_j)$,

(iv) conservation, $\sum_{j \neq i} b_K(\phi_j, \phi_i) = -b_K(\phi_i, \phi_i)$,

where $\mathcal{I}(K) := \{i : |supp(\phi_i) \cap K| \neq 0\}$.

Remark 3.3.2. *In the above definition, the 2nd property says that $-b_K(\phi_j, \phi_i)$ is a positive constant multiplication with $|K|$ if $\mathbf{a}_i$ and $\mathbf{a}_j$ are adjacent. The 4th property is related to the conservation of the numerical scheme, see Theorem 3.3.5. The 3rd property is proposed to make it easy to get zero row sum and zero column sum at the same time. One way to customize b_K is to define $b_K(\phi_j, \phi_i)$ first for all pairs with $j \neq i$, and then give a particular value to $b_K(\phi_i, \phi_i)$ to make the 4th property holds.*

Recall that the traditional way to introduce numerical viscosity depends on the use of the bilinear form corresponding to the Laplacian operator $-\nabla\cdot(\nu\nabla\psi)$. Here we use a bilinear form $B(\cdot,\cdot)$ to serve this purpose in a general sense, which is defined as the sum of bilinear forms $\nu_K b_K(\cdot,\cdot)$, where ν_K to be determined later is positive constant. It follows that

$$B(u_h, v_h) = \sum_{i,j} \sum_{K} U_i V_j \nu_K b_K(\phi_i, \phi_j) \tag{3.8}$$

for $u_h = \sum U_i \phi_i$ and $v_h = \sum V_i \phi_i$.

The use of the bilinear form B to introduce numerical viscosity is similar to the strategies used in other methods such as Streamline-upwind/Galerkin method, space-time Galerkin/least-squared method, subgrid scale method, characteristic Galerkin method and Taylor-Galerkin method, see (17) in [15].

Using (3.8) as artificial viscosity term, the semi-discretized approximation of the problem (3.1) is to find $u_h \in C^1([0,T]; V_h)$ such that

$$(\partial_t u_h, v_h)_L + (\nabla \cdot \mathbf{f}(u_h), v_h) + B(u_h, v_h) = 0, \quad \forall v_h \in V_h. \tag{3.9}$$

Using the forward Euler method for time stepping, we get a fully discrete method which is to find $u_h^{n+1} \in V_h$ as an approximation of the solution at time $t^{n+1} := t^n + \Delta t^n$ such that

$$(\frac{u_h^{n+1} - u_h^n}{\Delta t^n}, \phi_i)_L + (\nabla \cdot \mathbf{f}(u_h^n), \phi_i) + B(u_h^n, \phi_i) = 0, \quad \forall i, \tag{3.10}$$

i.e.,

$$(\frac{u_h^{n+1} - u_h^n}{\Delta t^n}, \phi_i)_L + (\nabla \cdot \mathbf{f}(u_h^n), \phi_i) + \sum_K \nu_K^n b_K(u_h^n, \phi_i) = 0, \quad \forall i, \tag{3.11}$$

where $u_h^n \in V_h$ is the given approximation at time t^n. The symbol ^{n} appears in ν_K^n because of its dependence on u_h^n. Using the definition of M_L and $m_i := \int \phi_i \, \mathrm{d}\mathbf{x}$, the algorithm (3.10) is expressed as

$$U_i^{n+1} = U_i^n - \frac{\Delta t^n}{m_i}[(\nabla \cdot \mathbf{f}(u_h^n), \phi_i) + \sum_K \nu_K^n b_K(u_h^n, \phi_i)], \quad \forall i. \tag{3.12}$$

Remark 3.3.3. *One reason to study the forward Euler method* (3.10) *is that it can be extended to any high-order Strong Stability Preserving Runger-Kutta (SSPRK) methods (see e.g., [33, 34]) which are convex combinations of the forward Euler method. This key property implies that the maximum principle property obtained for the forward Euler method holds also for SSPRK methods. For example, the two-stage second order Runge-*

Kutta method (SSPRK2) is expressed as

$$\begin{cases} u^{(1)} = & [u^n + \Delta t L(u^n)] \\ u^{n+1} = & \frac{1}{2}u^n + \frac{1}{2}[u^{(1)} + \Delta t L(u^{(1)})], \end{cases} \tag{3.13}$$

and three-stage third order Runge-Kutta method (SSPRK3) as

$$\begin{cases} u^{(1)} = & [u^n + \Delta t L(u^n)] \\ u^{(2)} = & \frac{3}{4}u^n + \frac{1}{4}[u^{(1)} + \Delta t L(u^{(1)})], \\ u^{n+1} = & \frac{1}{3}u^n + \frac{2}{3}[u^{(2)} + \Delta t L(u^{(2)})], \end{cases} \tag{3.14}$$

where the substeps using the forward Euler method are highlighted in brackets $[,]$.

Remark 3.3.4 (Time step in SPPRK methods)**.** *Note that the time step* Δt *obtained from the CFL condition based on* $u^{(i)}$ *may be different from the time step* Δt *based* $u^{(j)}$ *in general. Since* Δt *should be the same for all substeps of SSPRK methods, a loop has to be used in the numerical implementation, see for example Algorithm 1 where the constant* 0.9 *in Step-5 and Step-9 is user defined in order to minimize the number of iterations.*

Algorithm 1 One-step update of SSPRK3 algorithm (3.14)

Require: u_h^n
1: Estimate Δt_1 based on u_h^n.
2: Let $\Delta t := \Delta t_1$.
3: Get $u^{(1)}$ based on u_h^n using the forward Euler method with Δt.
4: Estimate Δt_2 based on $u_h^{(1)}$.
5: If $\Delta t_2 < \Delta t$, choose $\Delta t := 0.9\Delta t_2$, go to step-3.
6: Get $\tilde{u}^{(2)}$ based on $u^{(1)}$ using the forward Euler method with Δt.
7: Get $u^{(2)} = [3u_h^n + \tilde{u}^{(2)}]/4$.
8: Estimate Δt_3 based on $u_h^{(2)}$.
9: If $\Delta t_3 < \Delta t$, choose $\Delta t := 0.9\Delta t_3$, go to step-3.
10: Get $\tilde{u}^{(3)}$ based on $u^{(2)}$ using the forward Euler method with Δt.
11: **return** Get $u_h^{n+1} = [u_h^n + 2\tilde{u}^{(3)}]/3$.

Theorem 3.3.5. *The algorithm* (3.10) *is conservative, i.e.,*

$$\int_\Omega u_h^n \,\mathrm{d}\mathbf{x} = \int_\Omega u_h^0 \,\mathrm{d}\mathbf{x}, \quad \forall n, \tag{3.15}$$

provided that B satisfy the property that the sum of each column is 0, i.e.,

$$B(\phi_j, 1) = \sum_i B(\phi_j, \phi_i) = 0. \tag{3.16}$$

Proof. For fixed n, from (3.12), we have

$$\begin{aligned}
\int_\Omega u_h^{n+1} &= \int_\Omega \sum_j U_{h,j}^{n+1} \phi_j(x) \\
&= \sum_j U_{h,j}^{n+1} m_j \\
&= \sum_j (u_h^{n+1}, \phi_j)_L \\
&= \sum_j (u^n, \phi_j)_L - \sum_j \Delta t^n (\nabla \cdot \mathbf{f}(u_h^n), \phi_j) - \sum_j -\Delta t^n B(u^n, \phi_j) \\
&= \int_\Omega u^n - \Delta t^n \int_\Omega \nabla \cdot \mathbf{f}(u_h^n) - \Delta t^n B(u^n, 1) \\
&= \int_\Omega u^n - \Delta t^n \int_{\partial\Omega} \mathbf{n} \cdot \mathbf{f}(u_h^n) \\
&= \int_\Omega u_h^n,
\end{aligned}$$

where $\sum_j \phi(x) = 1$, $B(u^n, 1) = 0$ and the assumption on boundary condition are used. It follows that the scheme (3.10) is conservative.

□

Theorem 3.3.6. *Assume ν_K^n is chosen as follows*

$$\nu_K^n := \max_{\substack{i \neq j \\ i,j \in \mathcal{I}(K)}} \frac{[(\mathbf{f}'(u_h^n) \cdot \nabla \phi_j, \phi_i)_K]^+}{-b_K(\phi_j, \phi_i)}, \tag{3.17}$$

If Δt^n satisfies that

$$\Delta t^n \le \min_i \frac{m_i}{[(\mathbf{f}'(u_h^n)\cdot\nabla\phi_i,\phi_i)+\sum_K \nu_K^n b_K(\phi_i,\phi_i)]^+}, \tag{3.18}$$

then the solution u_h^{n+1} of (3.10) satisfies the following local maximum principle

$$\min_{j\in\mathcal{I}(S_i)} U_{h,i}^n \le U_i^{n+1} \le \max_{j\in\mathcal{I}(S_i)} U_{h,i}^n, \quad \forall i. \tag{3.19}$$

Proof. For fixed n, to ease notation, denote $u_h^{n+1} := \sum U^i\phi_i$ and $u_h^n := \sum U_i\phi_i$. By (3.12), since $\nabla\cdot\mathbf{f}(u_h^n) = \mathbf{f}'(u_h^n)\cdot\nabla u_h^n = \sum_j \mathbf{f}'(u_h^n)\cdot\nabla\phi_j U_j$ and b_K is bilinear, it follows that

$$U^i = U_i - \frac{\Delta t^n}{m_i}\sum_j[(\mathbf{f}'(u_h^n)\cdot\nabla\phi_j,\phi_i)+\sum_K \nu_K^n b_K(\phi_j,\phi_i)]U_j.$$

That is

$$\begin{aligned}
U^i =& \left\{1-\frac{\Delta t^n}{m_i}[(\mathbf{f}'(u_h^n)\cdot\nabla\phi_i,\phi_i)+\sum_K \nu_K^n b_K(\phi_i,\phi_i)]\right\}U_i \\
&+\frac{\Delta t^n}{m_i}\sum_{j\neq i}\left[-(\mathbf{f}'(u_h^n)\cdot\nabla\phi_j,\phi_i)-\sum_K \nu_K^n b_K(\phi_j,\phi_i)\right]U_j \\
=& \left\{1-\frac{\Delta t^n}{m_i}[(\mathbf{f}'(u_h^n)\cdot\nabla\phi_i,\phi_i)+\sum_K \nu_K^n b_K(\phi_i,\phi_i)]\right\}U_i \\
&+\frac{\Delta t^n}{m_i}\sum_{j\neq i}\sum_{K\subset S_{ij}}\left[-(\mathbf{f}'(u_h^n)\cdot\nabla\phi_j,\phi_i)_K-\nu_K^n b_K(\phi_j,\phi_i)\right]U_j \\
=&\alpha_i U_i+\sum_{j\neq i}\sum_{K\subset S_{ij}}\beta_{i,j,K}U_j,
\end{aligned}$$

where

$$\alpha_i := 1-\frac{\Delta t^n}{m_i}\left[(\mathbf{f}'(u_h^n)\cdot\nabla\phi_i,\phi_i)+\sum_K \nu_K^n b_K(\phi_i,\phi_i)\right]$$

and

$$\beta_{i,j,K} := -(\mathbf{f}'(u_h^n)\cdot\nabla\phi_j,\phi_i)_K-\nu_K^n b_K(\phi_j,\phi_i), \quad \forall j\neq i, K\subset S_{ij}.$$

Be definition of ν_K in (3.17), it follows that $\beta_{i,j,K} \geq 0$. Since

$$\begin{aligned}\alpha_i + \sum_{j\neq i}\sum_{K\subset S_{ij}} \beta_{i,j,K} =& 1 - \frac{\Delta t^n}{m_i}\sum_j [(\mathbf{f}'(u_h^n)\cdot\nabla\phi_j, \phi_i) + \sum_K \nu_K^n b_K(\phi_j,\phi_i)] \\ =& 1 - \frac{\Delta t^n}{m_i}[(\mathbf{f}'(u_h^n)\cdot\nabla\left(\sum_j \phi_j\right), \phi_i) + \sum_K \nu_K^n \sum_j b_K(\phi_j,\phi_i)],\end{aligned}$$

$\sum_j \phi_j = 1$ and $\sum_j b_K(\phi_j,\phi_i) = 0$, we obtain that U^i is a convex combination of $U_j, j \in \mathcal{I}(S_i)$ provided that $\alpha_i > 0$. The condition (3.18) implies that $\alpha_j > 0$, which completes the proof. □

Remark 3.3.7. *From the proof of the above theorem, it is readily to see that the numerical implementation of the 2nd term of* (3.10) *should be* $(\mathbf{f}'(u_h^n)\cdot\nabla u_h^n, \phi_i)$. *Additionally, the quadrature rule to compute it should be the same as the quadrature rule used in the integral used in the numerator of* ν_K^n *given in* (3.17).

Remark 3.3.8. *Note that in [36],* $\tilde{\nu}_K^n$ *is defined by*

$$\tilde{\nu}_K^n = \max_{\substack{i\neq j \\ i,j\in\mathcal{I}(K)}} \frac{[(\mathbf{f}'(u_h^n)\cdot\nabla\phi_j,\phi_i)_{S_{ij}}]^+}{\sum_{K\subset S_{ij}} -b_K(\phi_j,\phi_i)}. \tag{3.20}$$

Compared to ν_K^n *defined in* (3.17), *since it is not true that* $\nu_K^n \leq \tilde{\nu}_K^n$ *or* $\tilde{\nu}_K^n \leq \nu_K^n$, *maybe a better choice is*

$$\bar{\nu}_K^n = \max_{\substack{i\neq j \\ i,j\in\mathcal{I}(K)}} \min\left\{\frac{[(\mathbf{f}'(u_h^n)\cdot\nabla\phi_j,\phi_i)_K]^+}{-b_K(\phi_j,\phi_i)}, \frac{[(\mathbf{f}'(u_h^n)\cdot\nabla\phi_j,\phi_i)_{S_{ij}}]^+}{\sum_{K\subset S_{ij}} -b_K(\phi_j,\phi_i)}\right\}. \tag{3.21}$$

Remark 3.3.9. *In serial programming, the computation of* (3.20) *is more efficiently. The matrix* $[B]_{ij} := b_K(\phi_j,\phi_i)$ *can be assembled at the beginning. The matrix* $[C]_{ij} := (\mathbf{f}'(u_h^n)\cdot\nabla\phi_j,\phi_i)_{S_{ij}}$ *updated at the beginning of each time step for the flux term can be also used to get* (3.20). *However, the computation of* (3.17) *needs one more time numerical integral over each cell using the same quadrature rule. But, in parallel programming,* (3.17) *is easier to parallel since it involves only computations over a cell* K, *because using* (3.20)

needs more information from the neighboring cells which may be "ghost cells" on other processors. For comparison, please refer to the Method-1 and Method-2 in Section 3.9.

Lemma 3.3.10. *Assume there exists c_1 and c_2 such that*

$$c_1 \max_i \int_K \phi_i \le -b_K(\phi_i, \phi_j) \le c_2 \max_i \int_K \phi_i, \quad \forall i, j \in \mathcal{I}(K), i \neq j, \forall K. \tag{3.22}$$

If Δt^n satisfies that

$$\frac{\Delta t^n \beta_n}{\underline{h}} \le \frac{c_1}{c_1 + c_2}, \tag{3.23}$$

where

$$\beta_n := \max_{u \in [\min u_h^n, \max u_h^n]} \|\mathbf{f}'(u)\|_{l^2}, \tag{3.24}$$

then (3.18) *holds.*

Proof. By the definition ν_K in (3.17), the assumption (3.22) and the fact that $\phi_i \ge 0$, it follows

$$\nu_K^n \le \frac{\beta_n \int_K \phi_i(\mathbf{x}) \|\nabla \phi_j(\mathbf{x})\|_{l^2} \, \mathrm{d}\mathbf{x}}{c_1 \max_i \int_K \phi_i} \le \frac{\beta_n \int_K \phi_i}{h_K c_1 \max_i \int_K \phi_i} \le \frac{\beta_n}{h_K c_1}. \tag{3.25}$$

Since $\sum_j b_K(\phi_j, \phi_i) = 0$, the assumption (3.22) implies that

$$b_K(\phi_i, \phi_i) = -\sum_{j \neq i} b_K(\phi_j, \phi_i) \le (\mathcal{N}_K - 1) c_2 \max_i \int_K \phi_i, \quad K \subset S_i, \tag{3.26}$$

where $\mathcal{N}_K$ is the number of vertices of K. Combining (3.26) and (3.26) implies that

$$-\nu_K^n b_K(\phi_i, \phi_i) \le \frac{\beta_n c_2}{h_K c_1} \max_i \int_K \phi_i. \tag{3.27}$$

Since for any $K \subset S_i$

$$(\mathbf{f}'(u_h^n) \cdot \nabla \phi_i, \phi_i)_K \le \frac{\beta_n}{h_K} \int_K \phi_i, \tag{3.28}$$

we obtain that

$$\frac{\int_K \phi_i}{[(\mathbf{f}'(u_h^n)\cdot\nabla\phi_i,\phi_i)_K + \nu_K^n b_K(\phi_i,\phi_i)]^+} \geq \frac{\int_K \phi_i}{\frac{\beta_n}{h_K}\int_K \phi_i + \frac{\beta_n c_2}{h_K c_1}\max_i \int_K \phi_i} \geq \frac{h_K c_1}{\beta_n(c_1+c_2)},$$

which implies that

$$\frac{m_i}{[(\mathbf{f}'(u_h^n)\cdot\nabla\phi_i,\phi_i) - \sum_K \nu_K^n b_K(\phi_i,\phi_i)]^+} \geq \frac{\underline{h} c_1}{\beta_n(c_1+c_2)}.$$

Therefore, if Δt satisfies (3.23),then

$$\Delta t^n \leq \frac{m_i}{(\mathbf{f}'(u_h^n)\cdot\nabla\phi_i,\phi_i) - \sum_K \nu_K^n b_K(\phi_i,\phi_i)},$$

which is exactly the condition (3.18). □

Lemma 3.3.11. *The condition* (3.23) *is implied by the condition*

$$\frac{\Delta t^n \beta_0}{\underline{h}} \leq \frac{c_1}{c_1+c_2}, \tag{3.29}$$

where β_n *is defined in* (3.24).

Proof. Applying the above condition for Δt^0, by Theorem 3.3.6, we obtain that

$$\min_i U_i^0 \leq \min_i U_i^1 \leq \max_i U_i^1 \leq \max_i U_i^0,$$

which implies that

$$[\min u_h^1, \max u_h^1] \subset [\min u_h^0, \max u_h^0],$$

and hence $\beta_1 \leq \beta_0$. Inductively, we have that $\beta_n \leq \beta_0$. Therefore, the condition (3.29) implies the condition (3.23). □

Remark 3.3.12. *If b_K is chosen in [36]*

$$b_K(\phi_j,\phi_i)=\begin{cases}-\frac{1}{\mathcal{N}_K-1}|K|, & \text{if } i\neq j, i,j\in\mathcal{I}(K)\\ |K|, & \text{if } i=j,\\ 0, & \text{if } i\notin\mathcal{I}(K) \text{ or } j\notin\mathcal{I}(K),\end{cases} \tag{3.30}$$

which satisfying the Definition 3.3.1, then c_1 and c_2 defined in (3.22) *can be chosen as $c_1=c_2=\frac{1}{\mathcal{N}_K-1}$. It implies that $\frac{1}{2}$ can be chosen in the CFL condition* (3.29).

Remark 3.3.13 (Relation to graph Laplacian)**.** *Let K be a triangle with 3 vertices $\mathbf{a}_{\mathrm{j}^{\mathrm{geo}}(K,j)}, j=1,2,3$. The graph Laplacian L_K (see, e.g., [32, p. 286]) can be obtained when K is treated as a weighted graph with weight $\frac{|K|}{2}$, which satisfies that*

$$(L_K U,V)=\sum_{i=1,2,3}\sum_{j>i}\frac{|K|}{2}(U_{\mathrm{j}^{\mathrm{geo}}(K,i)}-U_{\mathrm{j}^{\mathrm{geo}}(K,j)})(V_{\mathrm{j}^{\mathrm{geo}}(K,i)}-V_{\mathrm{j}^{\mathrm{geo}}(K,j)}) \tag{3.31}$$

for any vector $U,V\in\mathbb{R}^{N^{\mathrm{geo}}}$. This equality can be obtained by adding 3 edges one by one. For example, assume the only nonzero weight is ω_{12} on the edge connecting $\mathbf{a}_{\mathrm{j}^{\mathrm{geo}}(K,1)}$ and $\mathbf{a}_{\mathrm{j}^{\mathrm{geo}}(K,2)}$. Denote this graph as K_1. Then from the definition of graph Laplacian in [32, p. 286], we get that

$$(L_{K_1}U,V)=\omega_{12}(U_{\mathrm{j}^{\mathrm{geo}}(K,1)}-U_{\mathrm{j}^{\mathrm{geo}}(K,2)})(V_{\mathrm{j}^{\mathrm{geo}}(K,1)}-V_{\mathrm{j}^{\mathrm{geo}}(K,2)}). \tag{3.32}$$

From the definition of b_K in (3.30)*, it is readily seen that*

$$b_K(u_h,v_h)=(L_K U,V) \tag{3.33}$$

for $u_h=\sum_i U_i\phi_i$ and $v_h=\sum_i V_i\phi_i$.

3.4 Backward Euler method

Compared to (3.10), the backward Euler method can also be constructed to preserve the maximum principle. The backward Euler method is to find $u_h^{n+1} \in V_h$ such that

$$\left(\frac{u_h^{n+1}-u_h^n}{\Delta t^n}, \phi_i\right)_L + (\nabla\cdot\mathbf{f}(u_h^{n+1}), \phi_i) + \sum_K \nu_K^n b_K(u_h^{n+1}, \phi_i) = 0, \quad \forall i, \tag{3.34}$$

Since the flux $\mathbf{f}(u)$ is usually nonlinear in u, we replace the 2nd term and get the following implicit algorithm

$$\left(\frac{u_h^{n+1}-u_h^n}{\Delta t^n}, \phi_i\right)_L + (\mathbf{f}'(u_h^n)\cdot\nabla u_h^{n+1}, \phi_i) + \sum_K \nu_K^n b_K(u_h^{n+1}, \phi_i) = 0, \quad \forall i. \tag{3.35}$$

The surprising thing is that ν_K^n defined in (3.17) also allows the above implicit algorithm to preserve the maximum principle (as a consequence of the following Theorem). Moreover, there is no constraint on time step Δt^n.

Theorem 3.4.1. *Assume ν_K^n is chosen as* (3.17)*. For any $\Delta t^n > 0$, the solution u_h^{n+1} of* (3.35) *has the following property*

(i) if $U_i^{n+1} = \max_{j\in\mathcal{I}(S_i)} U_j^{n+1}$, then

$$U_i^{n+1} \le \max_{j\in\mathcal{I}(S_i)} U_j^n; \tag{3.36}$$

(ii) if $U_i^{n+1} = \min_{j\in\mathcal{I}(S_i)} U_j^{n+1}$, then

$$U_i^{n+1} \ge \min_{j\in\mathcal{I}(S_i)} U_j^n. \tag{3.37}$$

Proof. For fixed n, to ease notation, denote $u_h^{n+1} := \sum U^i\phi_i$ and $u_h^n := \sum U_i\phi_i$. Using the

definition of M_L and $m_i := \int \phi_i \, \mathrm{d}\mathbf{x}$, the algorithm (3.35) is expressed as

$$U^i + \frac{\Delta t^n}{m_i} \sum_j \left[(\mathbf{f}'(u_h^n) \cdot \nabla \phi_j, \phi_i) + \sum_K \nu_K^n b_K(\phi_j, \phi_i) \right] U^j = U_i.$$

That is

$$\left\{ 1 + \frac{\Delta t^n}{m_i} [(\mathbf{f}'(u_h^n) \cdot \nabla \phi_i, \phi_i) + \sum_K \nu_K^n b_K(\phi_i, \phi_i)] \right\} U^i + \frac{\Delta t^n}{m_i} \sum_{j \neq i} [(\mathbf{f}'(u_h^n) \cdot \nabla \phi_j, \phi_i) + \sum_K \nu_K^n b_K(\phi_j, \phi_i)] U^j = U_i.$$

Define $\alpha_i := 1 + \frac{\Delta t^n}{m_i} [(\mathbf{f}'(u_h^n) \cdot \nabla \phi_i, \phi_i) + \sum_K \nu_K^n b_K(\phi_i, \phi_i)]$ and $\beta_{i,j,K} := -(\mathbf{f}'(u_h^n) \cdot \nabla \phi_j, \phi_i)_K - \nu_K^n b_K(\phi_j, \phi_i)$ for any $K \subset S_{ij}$. It follows that

$$\alpha_i U^i - \sum_{j \neq i} \sum_{K \subset S_{ij}} \beta_{i,j,K} U^j = U_i. \tag{3.38}$$

By definition of ν_K^n in (3.17), we get that $\beta_{i,j,K}$ is nonnegative. In fact, α_i is also positive for any Δt^n. This is because

$$\begin{aligned} (\mathbf{f}'(u_h^n) \cdot \nabla \phi_i, \phi_i) + \sum_K \nu_K^n b_K(\phi_i, \phi_i) &= \sum_{K \subset S_i} [(\mathbf{f}'(u_h^n) \cdot \nabla \phi_i, \phi_i)_K + \nu_K^n b_K(\phi_i, \phi_i)] \\ &= \sum_{K \subset S_i} \sum_{\substack{j \in \mathcal{I}(K) \\ j \neq i}} [-(\mathbf{f}'(u_h^n) \cdot \nabla \phi_j, \phi_i)_K - \nu_K^n b_K(\phi_j, \phi_i)] \\ &= \sum_{K \subset S_i} \sum_{\substack{j \in \mathcal{I}(K) \\ j \neq i}} \beta_{i,j,K} \geq 0, \end{aligned}$$

where in the 2nd equality we use the fact that $\sum_{j \in \mathcal{I}(K)} \phi_j(\mathbf{x}) = 1$ for $\forall \mathbf{x} \in K$ and $\sum_{j \in \mathcal{I}(K)} b_K(\phi_j, \phi_i) = 0$, for $\forall K$. Likewise, we have

$$\alpha_i - \sum_{j \neq i} \sum_{K \subset S_{ij}} \beta_{i,j,K} = 1.$$

(i) Assume $U^i = \max_{j\in\mathcal{I}(S_i)} U^j$. By (3.38), it follows that

$$\alpha_i U^i = U_i + \sum_{j\neq i}\sum_{K\subset S_{ij}} \beta_{i,j,K} U^j \leq \max_{j\in\mathcal{I}(S_i)} U_j + \sum_{j\neq i}\sum_{K\subset S_{ij}} \beta_{i,j,K} U^i,$$

and hence

$$[\alpha_i - \sum_{j\neq i}\sum_{K\subset S_{ij}} \beta_{i,j,K}]U^i \leq \max_{j\in\mathcal{I}(S_i)} U_j.$$

Since $\alpha_i - \sum_{j\neq i}\sum_{K\subset S_{ij}} = 1$, we obtain that

$$U^i \leq \max_{j\in\mathcal{I}(S_i)} U_j.$$

(ii) Assume $U^i = \min_{j\in\mathcal{I}(S_i)} U^j$. By (3.38), it follows that

$$\alpha_i U^i = U_i + \sum_{j\neq i}\sum_{K\subset S_{ij}} \beta_{i,j,K} U \geq \min_{j\in\mathcal{I}(S_i)} U_j + \sum_{j\neq i}\sum_{K\subset S_{ij}} \beta_{i,j,K} U^i,$$

and hence

$$[\alpha_i - \sum_{j\neq i}\sum_{K\subset S_{ij}} \beta_{i,j,K}]U^i \leq \min_{j\in\mathcal{I}(S_i)} U_j.$$

Since $\alpha_i - \sum_{j\neq i}\sum_{K\subset S_{ij}} \beta_{i,j,K} = 1$, we obtain that

$$U^i \leq \min_j U_j.$$

□

Remark 3.4.2. *The conclusion of Theorem 3.4 is not the same as the usual Local Extremum Diminishing property (LED). This is because the assumption that U_i^{n+1} is the local extremum does not imply that U_i^n is also the local extremum. For example, in one time step, the numerical solution $[U_1^n, U_2^n, U_3^n, U_4^n] = [1, 2, 1, 4]$ may become $[U_1^{n+1}, U_2^{n+1}, U_3^{n+1}, U_4^{n+1}]$*

$= [1, 2.5, 3.5, 1]$. *Although* $U_3^{n+1} = \max\{U_2^{n+1}, U_3^{n+1}, U_4^{n+1}\} \leq \max\{U_2^n, U_3^n, U_4^n\}$, *the local maximum* U_2^n *is increased.*

Applying Theorem 3.4 and neglecting the boundary issues, we obtain the following conclusion.

Corollary 3.4.3. *If* ν_K^n *is chosen as* (3.17)*, For any* $\Delta t^n > 0$*, the solution* u_h^{n+1} *of* (3.35) *satisfies the following global maximum principle:*

$$\min_i U_i^n \leq \min_i U_i^{n+1} \leq \max_i U_i^{n+1} \leq \max_i U_i^n, \quad \forall n.$$

Remark 3.4.4. *Note that even there are not restrictions on* Δt^n *for the implicit algorithm* (3.35) *to keep global maximum principle, we need to solve a linear system. If one modifies the small oscillations near the extrema of the solution in order to preserve the maximum principle exactly, then more techniques have to be used to keep the conservation property.*

3.5 High order method

One useful technique to get high-order method is to use the notion of entropy viscosity introduced in [38][40][36]. An inspiring example is the Riemann problem of Burgers equation $u_t(x,t) + (\frac{1}{2}u^2)_x = 0$ with initial data $u_0(x) = 1 - H(x)$, where $H(x)$ is the Heaviside function. The entropy solution has only one shock and is expressed as $u(x,t) = 1 - H(x - \frac{t}{2})$. Using delta function, we get that the entropy residual $(u^2)_t + (\frac{1}{3}u^3)_x = \frac{1}{2}\delta(x - \frac{t}{2}) - \frac{1}{6}\delta(x - \frac{t}{2}) = \frac{1}{3}\delta(x - \frac{t}{2})$, while the PDE residual is 0. Therefore, the entropy residual is a good indicator of the shock region. It can be used to choose a smaller numerical viscosity in the region far from the shock region in order to decrease the influence of numerical viscosity in the smooth region.

As in [37], choosing a convex entropy function $E \in Lip(\mathbb{R}, \mathbb{R})$ one can get a high order method by replacing $\nu_K^{L,n}$ introduced in (3.17) or (3.20) (the superscript L in $\nu_K^{L,n}$ stands

for the low order method) with $\nu_K^{H,n}$

$$\left(\frac{u_h^{n+1}-u_h^n}{\Delta t^n},\phi_i\right)_L+(\nabla\cdot\mathbf{f}(u_h^n),\phi_i)+\sum_K \nu_K^{H,n} b_K(u_h^n,\phi_i)=0,\quad \forall i, \tag{3.39}$$

where $\nu_K^{H,n}$ is defined by

$$\nu_K^{H,n}=\min\left(\nu_K^{L,n},\frac{c_E R_K(\tilde{u}_h^n,u_h^n)}{\|E(u_h^n)-\overline{E}(u_h^n)\|_{L^\infty(\Omega)}}\right), \tag{3.40}$$

where c_E is a user-defined parameter, $\overline{E}(u_h^n)$ is the mean of $E(u_h^n)$ over Ω, $R_K(\tilde{u}_h^n,u_h^n)$ is the entropy residual over K defined by

$$R_K(\tilde{u}_h^n,u_h^n)=\|\tfrac{1}{\Delta t^n}(E(\tilde{u}_h^n)-E(u_h^n))+\mathbf{f}'(u_h^n)\cdot\nabla E(u_h^n)\|_{L^\infty(K)}.$$

and $\tilde{u}_h^n$ is the pure Galerkin prediction of u_h^{n+1} without numerical viscosity given as follows

$$\left(\frac{\tilde{u}_h^n-u_h^n}{\Delta t^n},\phi_i\right)_L+(\nabla\cdot\mathbf{f}(u_h^n),\phi_i)=0,\quad \forall i. \tag{3.41}$$

3.6 Zalesak limiter

In numerical tests, we see that the algorithm (3.39) has high order accuracy. Some spurious oscillations appear at shock regions and the maximum principle is violated. As the authors stated in [61], one unavoidable difficulty in numerical computations of discontinuous solutions of conservation laws is that either the method is first-order accurate on smooth flows and the discontinuities are excessively smeared, or else spurious oscillations are introduced which pollute the solution and sometimes lead to nonlinear instability.

An interesting question is how to keep maximum principle and get high order convergence at the same time. One common technique of obtaining high resolution, second-order, oscillation free schemes is to use limiters [70]; In fact, all TVD schemes use limiter technique (see, e.g., [31, p. 169]).

The Zalesak limiter is used here to combine two schemes proposed in [36] and [40]

to obtain an explicit second-order maximum principle preserving numerical method that works on arbitrary meshes in any space dimension with any Lipschitz flux using continuous Lagrange finite elements. The Zalesak limiter (see, e.g., [80, 59, 58], [55, p. 52]), as an extension of the Flux Corrected Transport (FCT) method proposed by Boris and Book [5], is a two-step procedure based on the application of a low order scheme supplemented by the addition of a "limited" or "corrected" flux which is the difference between the flux of the high order scheme and that of the low order scheme.

One reason to choose the Zalesak limiter is that it is independent of the dimension of the problem. In contrast, if using other limiters like minmod, superbee, etc., then one needs to use direction splitting (since those limiters are inherently 1D), and find a way to compute the consecutive gradients which is not easy for unstructured mesh.

Assume there are two continuous finite element methods to solve (3.1): the low-order method and the high-order method as follows for given $u_h^n \in V_h$,

$$\begin{cases} \left(\frac{u_L^{n+1}-u_h^n}{\Delta t^n}, \phi_i\right)_L + (\nabla\cdot\mathbf{f}(u_h^n), \phi_i) + B_L(u_h^n, \phi_i) = 0, \quad \forall i, \\ \left(\frac{u_H^{n+1}-u_h^n}{\Delta t^n}, \phi_i\right)_L + (\nabla\cdot\mathbf{f}(u_h^n), \phi_i) + B_H(u_h^n, \phi_i) = 0, \quad \forall i. \end{cases} \tag{3.42}$$

where the bilinear forms B_L and B_H are defined by

$$B_L(u_h^n, \phi_i) = \sum_K \nu_K^{L,n} b_K(u_h^n, \phi_i), \quad B_H(u_h^n, \phi_i) = \sum_K \nu_K^{H,n} b_K(u_h^n, \phi_i), \tag{3.43}$$

with b_K satisfies the four properties as stated in Definition 3.3.1, and $\nu_K^{L,n}$ and $\nu_K^{H,n}$ are defined in (3.17) or (3.20) and (3.40). Assume the low-order method gives u_L^{n+1} which satisfies the maximum principle, and the high-order method produce u_h^{n+1} with high order accuracy but maybe the maximum principle does not hold.

The purpose of Zalesak limiter is to create a solution u_h^{n+1} from u_L^{n+1} and u_H^{n+1} such that it satisfies the maximum principle or even the local maximum principle and has high order convergence at the same time. In practice, the high-order solution u_H^{n+1} does not need to be computed, see (3.51).

From (3.42), we have that

$$\left(\frac{u_H^{n+1}-u_L^{n+1}}{\Delta t^n},\phi_i\right)_L + B_H(u_h^n,\phi_i) - B_L(u_h^n,\phi_i) = 0, \quad \forall i. \tag{3.44}$$

Lemma 3.6.1. *Assume B_H and B_L in* (3.42) *satisfy the property that each column sum is 0, i.e.,*

$$B_H(\phi_j,1) = \sum_i B_H(\phi_j,\phi_i) = 0, \quad B_L(\phi_j,1) = \sum_i B_L(\phi_j,\phi_i) = 0, \quad \forall j. \tag{3.45}$$

Then there exists an antisymmetric matrix A such that (3.44) *becomes*

$$m_i U_{i,H}^{n+1} = m_i U_{L,i}^{n+1} + \Delta t^n \sum_{j\neq i} A_{ij}, \tag{3.46}$$

In particular, A can be chosen as

$$A_{ij} := -[B_H(\phi_j,\phi_i)U_j^n - B_H(\phi_i,\phi_j)U_i^n] + [B_L(\phi_j,\phi_i)U_j^n - B_L(\phi_i,\phi_j)U_i^n]. \tag{3.47}$$

Proof. Using the definition of m_i and mass lumping, from (3.44), we obtain that

$$m_i U_H^{n+1} - m_i U_L^{n+1} + \Delta t^n [B_H(u_h^n,\phi_i) - B_L(u_h^n,\phi_i)] = 0. \tag{3.48}$$

Since $\sum_i B_H(\phi_j,\phi_i) = 0$, we get

$$\begin{aligned} B_H(u_h^n,\phi_i) &= \sum_j B_H(\phi_j,\phi_i)U_j^n \\ &= \sum_{j\neq i} B_H(\phi_j,\phi_i)U_j^n + B_H(\phi_i,\phi_i)U_i^n \\ &= \sum_{j\neq i} [B_H(\phi_j,\phi_i)U_j^n - B_H(\phi_i,\phi_j)U_i^n]. \end{aligned}$$

Likewise, we have

$$B_L(u_h^n, \phi_i) = \sum_{j \neq i} [B_L(\phi_j, \phi_i) U_j^n - B_L(\phi_i, \phi_j) U_i^n].$$

Define

$$A_{ij} := -[B_H(\phi_j, \phi_i) U_j^n - B_H(\phi_i, \phi_j) U_i^n] + [B_L(\phi_j, \phi_i) U_j^n - B_L(\phi_i, \phi_j) U_i^n]. \tag{3.49}$$

It is readily seen that $A = [A_{ij}]$ is antisymmetric and gives (3.46). □

Corollary 3.6.2. *If the bilinear form B in* (3.42) *is symmetric, then A defined in* (3.47) *is equal to*

$$A_{ij} = [-B_H(\phi_i, \phi_j) + B_L(\phi_i, \phi_j)][U_j^n - U_i^n]. \tag{3.50}$$

Lemma 3.6.3. *The choice of A satisfying* (3.46) *is not unique.*

Proof. Choose 3 indexes arbitrarily. Without lose of generality, let us assume it is $\{1, 2, 3\}$. Define the matrix B to be

$$B = \begin{bmatrix} 0 & -1 & 1 & \dots & 0 \\ 1 & 0 & -1 & \dots & 0 \\ -1 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}_{N^{\text{geo}} \times N^{\text{geo}}}.$$

By direct computation, we can see that if A satisfies (3.46), so does $A + B$. □

Based on the idea of *Zalesak's limiter* (see, e.g., [80, 59, 58]), we will construct $u_h^{n+1} := \sum U_i^{n+1} \phi_i$ as an approximation solution at t^{n+1} with no new spurious unphysical over-

shoots and under-shoots

$$m_i U_i^{n+1} = m_i U_{L,i}^{n+1} + \sum_{j \neq i} \alpha_{ij} A_{ij}, \quad \forall i, \tag{3.51}$$

where the so-called "limiter" α, is a matrix obtained in the Algorithm 2.

Algorithm 2 Zalesak Limiter

Require: u^L and $[A_{ij}]$

1: **for** $i = 0$ **to** N **do**

2: Get $U_{L,i}^{n+1}$ and A_{ij}, $j = 1, \cdots, N$.

3: Compute

$$P_i^+ := \sum_{j \neq i} (A_{ij})^+, \quad P_i^- := \sum_{j \neq i} (A_{ij})^-. \tag{3.52}$$

4: Define

$$Q_i^+ := m_i (U_i^{\max} - U_{L,i}^{n+1}), \quad Q_i^- := m_i (U_{L,i}^{n+1} - U_i^{\min}),$$

where

$$U_i^{\max} := \max\{U_j^n, j \in \mathcal{I}(S_i)\}, \quad U_i^{\min} := \min\{U_j^n, j \in \mathcal{I}(S_i)\},$$

5: Evaluate

$$R_i^+ := \begin{cases} \min\{1, \frac{Q_i^+}{P_i^+}\} & \text{for} \quad P_i^+ \neq 0, \\ 1 & \text{for} \quad P_i^+ = 0. \end{cases} \qquad R_i^- := \begin{cases} \min\{1, \frac{Q_i^-}{P_i^-}\} & \text{for} \quad P_i^- \neq 0, \\ 1 & \text{for} \quad P_i^- = 0. \end{cases}$$

6: Choose

$$\alpha_{ij} = \begin{cases} \min\{R_i^+, R_j^-\} & \text{for} \quad A_{ij} \geq 0, \\ \min\{R_i^-, R_j^+\} & \text{for} \quad A_{ij} < 0. \end{cases} \tag{3.53}$$

7: Return $\alpha := [\alpha_{ij}]$.

8: **end for**

From Algorithm 2, we get the following properties.

Lemma 3.6.4.

$$P_i^+ \geq 0, \quad P_i^- \geq 0, \quad Q_i^+ \geq 0, \quad Q_i^- \geq 0, \quad R_i^+ \geq 0, \quad R_i^- \geq 0, \quad \forall\, i, \tag{3.54}$$

and $\quad 0 \leq \alpha_{ij} \leq 1$, *for* $\forall\, i, j$.

The algorithm (3.51) is conservative since α is symmetric, A is antisymmetric and the low order method is conservative.

Lemma 3.6.5. *If the low-order method is conservative, i.e.,*

$$\int_\Omega u_L^{n+1} = \int_\Omega u_h^n, \quad \forall n, \tag{3.55}$$

then the scheme (3.51) *is conservative.*

Proof. If $A_{ij} \geq 0$, from the definition of α_{ij}, we obtain that $\alpha_{ij} = \min\{R_i^+, R_j^-\}$ and at the same time $\alpha_{ji} = \min\{R_j^-, R_i^+\}$ since $A_{ji} = -A_{ij} \leq 0$ by Lemma 3.6.1. The case $A_{ij} \leq 0$ can be considered in the same way. It follows that $\alpha_{ij} = \alpha_{ji}$. Then we have

$$\begin{aligned}
\int_\Omega u_h^{n+1} &= \sum_i m_i U_i^{n+1} \\
&= \sum_i m_i U_{L,i}^n + \sum_i \sum_{j \neq i} \alpha_{ij} A_{ij} \\
&= \int_\Omega u_h^n + \sum_{i<j} (\alpha_{ij} A_{ij} + \alpha_{ji} A_{ji}) \\
&= \int_\Omega u_h^n + \sum_{i<j} (\alpha_{ij} A_{ij} - \alpha_{ij} A_{ij}) \\
&= \int_\Omega u_h^n,
\end{aligned}$$

where in the third equality we use the assumption that the low-order method is conservative. Therefore the scheme (3.51) is conservative. □

Furthermore, the scheme (3.51) satisfies the local maximum principle.

Theorem 3.6.6. *For any fixed n, if $u_{h,L}^{n+1}$ satisfies the local maximum principle, then the the solution u_h^{n+1} of the scheme* (3.51) *satisfies the same local maximum principle*

$$\min_{j\in\mathcal{I}(S_i)} U_j^n \le U_i^{n+1} \le \max_{j\in\mathcal{I}(S_i)} U_j^n, \quad \forall i. \tag{3.56}$$

Proof. By (3.51) and the definition of α_{ij} in (3.47), we obtain that for any i,

$$\begin{aligned}
U_i^{n+1} :=& U_{L,i}^{n+1} + \sum_{j\neq i} \alpha_{ij}\frac{A_{ij}}{m_i} \le U_{L,i}^{n+1} + \sum_{j\neq i, A_{ij}\ge 0} \alpha_{ij}\frac{A_{ij}}{m_i} \\
=& U_{L,i}^{n+1} + \sum_{j\neq i, A_{ij}\ge 0} \min\{R_i^+, R_j^-\}\frac{A_{ij}}{m_i} \\
\le& U_{L,i}^{n+1} + \sum_{j\neq i, A_{ij}\ge 0} R_i^+\frac{A_{ij}}{m_i} \\
\le& U_{L,i}^{n+1} + \sum_{j\neq i, A_{ij}\ge 0} \frac{Q_i^+}{P_i^+}\frac{A_{ij}}{m_i} \\
=& U_{L,i}^{n+1} + \frac{Q_i^+}{m_i}\frac{\sum_{j\neq i, A_{ij}\ge 0} A_{ij}}{P_i^+} = U_{L,i}^{n+1} + \frac{Q_i^+}{m_i} = U_i^{\max} \le \max_{j\in\mathcal{I}(S_i)} U_j^n.
\end{aligned}$$

Similarly for the lower bound, we have

$$\begin{aligned}
U_i^{n+1} :=& U_{L,i}^{n+1} + \sum_{j\neq i} \alpha_{ij}\frac{A_{ij}}{m_i} \ge U_{L,i}^{n+1} + \sum_{j\neq i, A_{ij}\le 0} \alpha_{ij}\frac{A_{ij}}{m_i} \\
=& U_{L,i}^{n+1} + \sum_{j\neq i, A_{ij}\le 0} \min\{R_i^-, R_j^+\}\frac{A_{ij}}{m_i} \\
\ge& U_{L,i}^{n+1} + \sum_{j\neq i, A_{ij}\le 0} R_i^-\frac{a_{ij}}{m_i} \\
\ge& U_{L,i}^{n+1} + \sum_{j\neq i, A_{ij}\le 0} \frac{Q_i^-}{p_i^-}\frac{a_{ij}}{m_i} \\
=& U_{L,i}^{n+1} + \frac{Q_i^-}{m_i}\frac{\sum_{j\neq i, A_{ij}\le 0} A_{ij}}{P_i^-} \\
=& U_{L,i}^{n+1} + \frac{Q_i^-}{m_i} = U_{L,i}^{n+1} + \sum_{j\neq i, A_{ij}\le 0} \frac{Q_i^-}{m_i} = U_i^{\min} \ge \min_{j\in\mathcal{I}(S_i)} U_j^n,
\end{aligned}$$

which completes the proof. □

Remark 3.6.7. *For other choice of $U_i^{\max}$ and $U_i^{\min}$, the above two properties still hold. For example, $U_i^{\max}$ and $U_i^{\min}$ can be chosen as*

$$U_i^{\max} := \max\{U_{L,j}^{n+1}, j \in \mathcal{I}(S_i)\},$$

and

$$U_i^{\min} := \min\{U_{L,j}^{n+1}, j \in \mathcal{I}(S_i)\},$$

since the low order method is assumed to satisfy the maximum principle.

The Theorem 3.6.6 implies that if the low order scheme satisfies the maximum principle, then so does the new scheme (3.51).

3.7 Generalized Zalesak limiter

As a generalization of the scheme (3.51) using Zalesak limiter, we propose the following new scheme to get

$$u_h^{n+1} := \sum_j U_j^{n+1}\phi_j$$

such that

$$m_i U_i^{n+1} = m_i U_{L,i}^{n+1} + \sum_j \tilde{A}_{ij}, \tag{3.57}$$

where $\tilde{A}$ is defined in Algorithm 3 based on A introduced in (3.47). Comparing to the scheme (3.51), one can see that the scheme (3.51) is more efficient.

The quantity Q_i^+, Q_i^- and $\tilde{A}_{ij}$ have the following property.

Lemma 3.7.1. $Q_i^+ \geq 0, \quad Q_i^- \geq 0, \quad \tilde{A}_{ij} = -\tilde{A}_{ji}.$

Proof. Since the low-order method is assumed to satisfy the maximum principle, we obtain that Q_i^+ and Q_i^- are positive from their definition in (3.58). By Lemma 3.6.1 A is antisymmetric. It follows that $\tilde{A}$ is antisymmetric from (3.59). □

Algorithm 3 Generalized Zalesak Limiter

Require: u^L and $[A_{ij}]$

1: **for** $i = 0$ **to** N **do**

2: Get $U_{L,i}^{n+1}$ and A_{ij}, $j = 1, \cdots, N$.

3: Define

$$\mathcal{I}_i^+ = \{j : A_{ij} > 0\}, \quad \mathcal{I}_i^- = \{j : A_{ij} < 0\}$$

4: Choose $\omega_{ij} (\geq 0)$ for each $A_{ij} \neq 0$ such that

$$\sum_{j \in \mathcal{I}_i^+} \omega_{ij} \leq Q_i^+, \quad \sum_{j \in \mathcal{I}_i^-} \omega_{ij} \leq Q_i^-$$

where

$$Q_i^+ := m_i (U_i^{\max} - U_{L,i}^{n+1}), \quad Q_i^- := m_i (U_{L,i}^{n+1} - U_i^{\min}), \tag{3.58}$$

and

$$U_i^{\max} := \max\{U_j^n, j \in \mathcal{I}(S_i)\}, \quad U_i^{\min} := \min\{U_j^n, j \in \mathcal{I}(S_i)\},$$

5: Loop each edge and define

$$\tilde{A}_{ij} = \operatorname{sgn}(A_{ij}) \min\{\omega_{ij}, \omega_{ji}\}. \tag{3.59}$$

6: Return $m_i U_i^{n+1} = m_i U_{L,i}^{n+1} + \sum_j \tilde{A}_{ij}$.

7: **end for**

Lemma 3.7.2. *If the low order method is conservative, i.e.,*

$$\sum_i m_i U_{L,i}^{n+1} = \int_\Omega u_h^n, \quad \forall n,$$

then the algorithm (3.57) *is also conservative.*

Proof. This is because

$$\begin{aligned}\int_\Omega u_h^{n+1} =& \sum_i m_i U_i^{n+1} \\ =& \sum_i m_i U_{L,i}^{n+1} + \sum_i \sum_{j\neq i} \tilde{A}_{ij} \\ =& \int_\Omega u_h^n + \sum_{i<j} (\tilde{A}_{ij} + \tilde{A}_{ji}) \\ =& \int_\Omega u_h^n,\end{aligned}$$

□

Theorem 3.7.3. *For any fixed n, if $u_{h,L}^{n+1}$ satisfies the local maximum principle, then the solution u_h^{n+1} of the scheme* (3.57) *satisfies the same local maximum principle*

$$\min_{j\in\mathcal{I}(S_i)} U_j^n \leq U_i^{n+1} \leq \max_{j\in\mathcal{I}(S_i)} U_j^n, \quad \forall i. \tag{3.60}$$

Proof. For any i, we have

$$\begin{aligned}U_i^{n+1} :=& U_{L,i}^{n+1} + \sum_{j\neq i} \frac{\tilde{A}_{ij}}{m_i} \\ \leq& U_{L,i}^{n+1} + \sum_{j\neq i, A_{ij}>0} \frac{\tilde{A}_{ij}}{m_i} \\ =& U_{L,i}^{n+1} + \sum_{j\in\mathcal{I}_i^+} \frac{\min\{\omega_{ij},\omega_{ji}\}}{m_i} \\ \leq& U_{L,i}^{n+1} + \sum_{j\in\mathcal{I}_i^+} \frac{\omega_{ij}}{m_i} \\ \leq& U_{L,i}^{n+1} + \frac{Q_i^+}{m_i} \\ =& U_i^{\max} \\ \leq& \max_{j\in\mathcal{I}(S_i)} U_j^n.\end{aligned}$$

Similarly for the lower bound, we have

$$\begin{aligned}
U_i^{n+1} :=& U_{L,i}^{n+1} + \sum_{j\neq i} \frac{\tilde{A}_{ij}}{m_i} \\
\geq& U_{L,i}^{n+1} - \sum_{j\neq i, A_{ij}<0} \frac{\tilde{A}_{ij}}{m_i} \\
=& U_{L,i}^{n+1} - \sum_{j\in\mathcal{I}_i^-} \frac{\min\{\omega_{ij},\omega_{ji}\}}{m_i} \\
\geq& U_{L,i}^{n+1} - \sum_{j\in\mathcal{I}_i^-} \frac{\omega_{ij}}{m_i} \\
\geq& U_{L,i}^{n+1} - \frac{Q_i^-}{m_i} \\
=& U_i^{\min} \\
\geq& \min_{j\in\mathcal{I}(S_i)} U_j^n.
\end{aligned}$$

□

The reason to call the algorithm (3.57) generalized Zalesak limiter is that for some special choice of ω_{ij}, it becomes the original Zalesak limiter.

Lemma 3.7.4. *If ω_{ij} is defined to be*

$$\omega_{ij} = \begin{cases} \min\left\{A_{ij}, A_{ij}\frac{Q_i^+}{P_i^+}\right\}, & \text{if } A_{ij} > 0, \\ \min\left\{-A_{ij}, -A_{ij}\frac{Q_i^-}{P_i^-}\right\}, & \text{if } A_{ij} < 0, \end{cases} \tag{3.61}$$

where P_i^+ and P_i^- are defined in (3.52)*, then the scheme* (3.57) *using generalized Zalesak limiter is equal to the scheme* (3.51) *using the Zalesak limiter.*

Proof. Comparing Algorithm 3 and Algorithm 2, we need to show that

$$\tilde{A}_{ij} = \alpha_{ij} A_{ij}. \tag{3.62}$$

Assume $A_{ij} \geq 0$. By the definition α_{ij} in (3.53), it follows that

$$\begin{aligned}
\alpha_{ij} A_{ij} &= \min\{R_i^+, R_j^-\} A_{ij} \\
&= \min\left\{1, \frac{Q_i^+}{P_i^+}, \frac{Q_j^-}{P_j^-}\right\} A_{ij} \\
&= \min\left\{A_{ij}, A_{ij}\frac{Q_i^+}{P_i^+}, A_{ij}\frac{Q_j^-}{P_j^-}\right\}.
\end{aligned}$$

By the definition $\tilde{A}_{ij}$ in (3.59), we have

$$\begin{aligned}
\tilde{A}_{ij} &= \min\{\omega_{ij}, \omega_{ji}\} \\
&= \min\left\{A_{ij}, A_{ij}\frac{Q_i^+}{P_i^+}, -A_{ji}\frac{Q_j^-}{P_j^-}\right\} \\
&= \min\left\{A_{ij}, A_{ij}\frac{Q_i^+}{P_i^+}, A_{ij}\frac{Q_j^-}{P_j^-}\right\},
\end{aligned}$$

which implies that $\tilde{A}_{ij} = \alpha_{ij} A_{ij}$. Assume $A_{ij} \leq 0$. By the definition α_{ij} in (3.53), it follows that

$$\begin{aligned}
\alpha_{ij} A_{ij} &= \min\{R_i^-, R_j^+\} A_{ij} \\
&= \min\left\{1, \frac{Q_i^-}{P_i^-}, \frac{Q_j^+}{P_j^+}\right\} A_{ij} \\
&= -\min\left\{A_{ji}, A_{ji}\frac{Q_i^-}{P_i^-}, A_{ji}\frac{Q_j^+}{P_j^+}\right\}.
\end{aligned}$$

By the definition $\tilde{A}_{ij}$ in (3.59), we have

$$\begin{aligned}
\tilde{A}_{ij} &= -\min\{\omega_{ij}, \omega_{ji}\} \\
&= -\min\left\{-A_{ij}, -A_{ij}\frac{Q_i^-}{P_i^-}, A_{ji}\frac{Q_j^+}{P_j^+}\right\} \\
&= -\min\left\{A_{ji}, A_{ji}\frac{Q_i^-}{P_i^-}, A_{ji}\frac{Q_j^+}{P_j^+}\right\},
\end{aligned}$$

which implies that $\tilde{A}_{ij} = \alpha_{ij} A_{ij}$. □

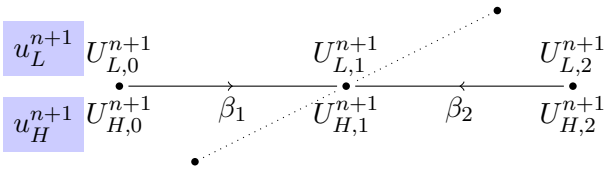
Remark 3.7.5. *One strategy to choose ω_{ij} is stated as follows. For fixed i,assume the order of all positive elements in i-row is $A_{ij_l} \geq \cdots \geq A_{ij_1} > 0$. Choose $m \in \{1, \cdots, l\}$ such that $\sum_{n=1,\cdots,m} A_{ij_n} \leq Q_i^+$. Then define $\omega_{ij_n} = A_{ij_n}, n = 1, \cdots, m$. For $n \in \{m+1, \cdots, l\}$, choose $\omega_{ij_n} = \min\{A_{ij_n}, A_{ij_n} \frac{Q_i^+ - \sum_{n=1,\cdots,m} A_{ij_n}}{\sum_{n=m+1,\cdots,l} A_{ij_n}}\}$ inspired by (3.61). Likewise, we can get ω_{ij} for $A_{ij} \leq 0$. The idea behind this strategy is to use small flux in high-order method to a large degree.*

Remark 3.7.6. *The scheme (3.57) is equivalent to the scheme (3.63) uby defining α_{ij} as*

$$\alpha_{ij} := \frac{\tilde{A}_{ij}}{A_{ij}}. \tag{3.63}$$

In order to make the limiter (3.63) in the bound $[0, 1]$, we can require $\omega_{ij} \leq A_{ij}$. This requirement is enforced for the original Zalesak limiter. See Lemma 3.7.4.

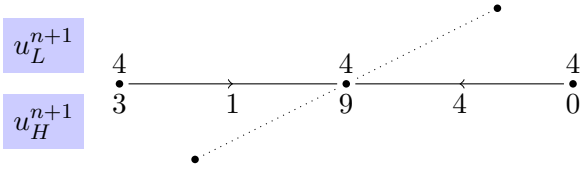
Example 3.7.7. *Consider a 1D problem. Let us use the following graph*



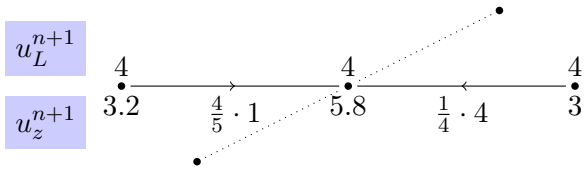
to show the relation between low-order solution and high-order solution where the arrow of the line between two nodes means $A_{10} = +\beta_1, A_{01} = -\beta_1, A_{12} = +\beta_2$,and $A_{21} = -\beta_2$. Considering Node-1 from the above graph, we see that

$$U_{H,1}^{n+1} = U_{L,1}^{n+1} + A_{10} + A_{12}. \tag{3.64}$$

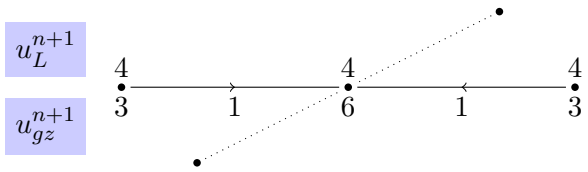
Assume the low-order method and the high-order method gives the following results



Assume $m_i = 1$, *and the local bounds* $[U_i^{\min}, U_i^{\max}]$ *at the three nodes* $i = 0, 1, 2$ *are* $[3, 5], [3, 8]$ *and* $[3, 5]$. *It follows that the high-order method violates the local maximum principle at nodes 1 and 2. Using Zalesak limiter* (3.51), *we get* $\alpha_{10} = \frac{4}{5}$ *and* $\alpha_{12} = \frac{1}{4}$



If we apply the generalized Zalesak limiter (3.57), *and assume* $\omega_{10} = 1$ *and* $\omega_{12} = 3$. *Since* $\omega_{01} = 1$ *and* $\omega_{21} = 1$, *i.e.,* $\alpha_{10} = 1$ *and* $\alpha_{12} = \frac{1}{4}$, *we get that following result*



3.8 Mass correction

As stated in the previous Sections, the Zalesak limiter and generalized Zalesak limiter can be used to combine the following two methods

$$\begin{cases} \left(\frac{u_L^{n+1}-u_h^n}{\Delta t^n}, \phi_i\right)_L + (\nabla\cdot\mathbf{f}(u_h^n), \phi_i) + B_L(u_h^n, \phi_i) = 0, \quad \forall i, \\ \left(\frac{u_H^{n+1}-u_h^n}{\Delta t^n}, \phi_i\right)_L + (\nabla\cdot\mathbf{f}(u_h^n), \phi_i) + B_H(u_h^n, \phi_i) = 0, \quad \forall i. \end{cases} \tag{3.65}$$

where the two bilinear forms B_L and B_H are defined as

$$B_L(u_h^n, \phi_i) = \sum_K \nu_K^{L,n} b_K(u_h^n, \phi_i), \quad B_H(u_h^n, \phi_i) = \sum_K \nu_K^{H,n} b_K(u_h^n, \phi_i), \tag{3.66}$$

with b_K satisfies the four properties as stated in Definition 3.3.1, and $\nu_K^{L,n}$ and $\nu_K^{H,n}$ are defined in (3.17) or (3.20) and (3.40).

It is well-known that lumping the mass matrix induces high dispersion errors as shown in [12, 13, 75, 39]. It is desirable to use M_C in high-order method. One interesting question

is how to combine it with the low order methods. Assume two methods are given as follows

$$\begin{cases} \left(\frac{u_L^{n+1}-u_h^n}{\Delta t^n}, \phi_i\right)_L + (\nabla\cdot\mathbf{f}(u_h^n), \phi_i) + B_L(u_h^n, \phi_i) = & 0, \quad \forall i, \\ \left(\frac{u_H^{n+1}-u_h^n}{\Delta t^n}, \phi_i\right)_H + (\nabla\cdot\mathbf{f}(u_h^n), \phi_i) + B_H(u_h^n, \phi_i) = & 0, \quad \forall i, \end{cases} \tag{3.67}$$

where the bilinear form $(\cdot,\cdot)_H : V_h \times V_h \to \mathbb{R}$ is defined by

$$(u, v)_H := U^{\mathsf{T}}(I - B)V,$$

with $u = \sum_i U_i\phi_i$, $U = (U_1, \cdots, U_{n^{\text{geo}}})^{\mathsf{T}}$ and B is defined in (3.8).

To get the discrete form of the low-order method, denoting by $F \in \mathbb{R}^{n^{\text{geo}}}$ the column vector with entries

$$F_i := (\nabla\cdot\mathbf{f}(u_h^n), \phi_i) + B_L(u_h^n, \phi_i) = \sum_{K\subset S_i}\left(\int_K \nabla\cdot(\mathbf{f}(u_h^n))\varphi_i \,\mathrm{d}\mathbf{x} + \nu_K^{L,n} b_K(u_h^n, \varphi_i)\right), \tag{3.68}$$

the definition of u_L^{n+1} gives

$$U_L^{n+1} = U^n - \Delta t^n (M_L)^{-1} F, \tag{3.69}$$

where M_L is the lumped mass matrix which is a diagonal matrix with i-th term $m_i = \int_\Omega \phi_i$.

Based on an idea from [39] we use the Neumann series to approximate the inverse of consistent mass matrix $M_C = [\int_\Omega \phi_i\phi_j]_{n^{\text{geo}}\times n^{\text{geo}}}$ by the following formula

$$M_C^{-1} = (M_L)^{-1}(I + B + B^2 + \ldots) \approx (M_L)^{-1}(I + B), \tag{3.70}$$

where matrix B is defined by

$$B := (M_L - M_C)(M_L)^{-1}. \tag{3.71}$$

The Neumann series can be shown to be convergent. For instance, it is shown in [39] that

the spectral radius of B is less than $\frac{3}{4}$ for $\mathbb{P}_1$ Lagrange finite elements in two dimensional space. Denoting by $G \in \mathbb{R}^{N^{\text{geo}}}$ the column vector with entries

$$G_i := (\nabla\cdot\mathbf{f}(u_h^n), \phi_i) + B_H(u_h^n, \phi_i) = \sum_{K\subset S_i} \left(\int_K \nabla\cdot(\mathbf{f}(u_h^n))\varphi_i \,\mathrm{d}\mathbf{x} + \nu_K^{H,n} b_K(u_h^n, \varphi_i) \right), \quad (3.72)$$

we then define the high-order solution

$$u_H^{n+1} := \sum_{i=1}^{N} U_{H,i}^{n+1} \varphi_i$$

as follows:

$$U_H^{n+1} = U^n - \Delta t^n (M_L)^{-1}(I + B)G. \quad (3.73)$$

Remark 3.8.1. *Although* (3.73) *is not equal to the discrete form of the high-order scheme in* (3.42)*, it is shown in [39] that approximating* $(M_C)^{-1}$ *by* $(M_L)^{-1}(I+B)$ *exactly corrects the dispersion error induced by mass lumping for* $\mathbb{P}_1$ *elements.*

Note that although M_C is symmetric, B is not. However, the sum of each column of B is 0.

Lemma 3.8.2. *The matrix* B *defined in* (3.8) *has the property that the sum of each column is 0, i.e.,*

$$\sum_i B_{ij} = 0, \quad \forall j. \quad (3.74)$$

Proof. Be definition of B (3.8), it follows that

$$\begin{aligned}
\sum_i B_{ij} =& \frac{1}{m_j} \sum_i ([M_L]_{ij} - [M_C]_{ij}) \\
=& \frac{1}{m_j} (m_j - \sum_i [M_C]_{ij}) \\
=& \frac{1}{m_j} (m_j - \sum_i \int_\Omega \phi_i \phi_j \,\mathrm{d}\mathbf{x}) \\
=& \frac{1}{m_j} (m_j - \int_\Omega \phi_j \,\mathrm{d}\mathbf{x}) = 0.
\end{aligned}$$

□

Lemma 3.8.3. *There exists an antisymmetric matrix A such that* (3.67) *becomes*

$$m_i U_{i,H}^{n+1} = m_i U_{L,i}^{n+1} + \Delta t^n \sum_{j \neq i} A_{ij}. \tag{3.75}$$

In particular, A can be chosen as follows

$$A_{ij} := [B_L(\phi_i, \phi_j) - B_H(\phi_i, \phi_j)][U_j^n - U_i^n] - [B_{ij} G_j - B_{ji} G_i]. \tag{3.76}$$

Proof. From (3.69) and (3.73), it follows that

$$\begin{aligned} U_H^{n+1} =& U_L^{n+1} + \Delta t^n (M_L)^{-1} F - \Delta t^n (M_L)^{-1} (I + B) G \\ =& U_L^{n+1} + \Delta t^n (M_L)^{-1} (F - G) - \Delta t^n (M_L)^{-1} B G. \end{aligned}$$

That is

$$m_i U_{H,i}^{n+1} = m_i U_{L,i}^{n+1} + \Delta t^n (F_i - G_i) - \Delta t^n (BG)_i.$$

Using the definition of F in (3.68) and the definition of Gin (3.72), we obtain that

$$F_i - G_i = B_L(u_h^n, \phi_i) - B_H(u_h^n, \phi_i).$$

In analogy with the proof of Lemma 3.6.1, the property that two bilinear forms B_L and B_H are symmetric and satisfy $B_L(\phi_i, 1) = B_H(\phi_i, 1) = 0$ implies that

$$\begin{aligned} F_i - G_i =& \sum_{j \neq i} [B_L(\phi_i, \phi_j) - B_H(\phi_i, \phi_j)][U_j^n - U_i^n] \\ =:& \sum_{j \neq i} A_{ij}^1, \end{aligned}$$

and the matrix $A^1 = [A^1_{ij}]$ is antisymmetric. Similarly, Lemma 3.8.2 implies that

$$\begin{aligned}(BG)_i =& \sum_j B_{ij}G_j \\ =& \sum_{j\neq i} B_{ij}G_j + B_{ii}G_i \\ =& \sum_{j\neq i}[B_{ij}G_j - B_{ji}G_i] =: \sum_{j\neq i} A^2_{ij}.\end{aligned}$$

Therefore, we conclude that $A := A^1 - A^2$ is antisymmetric and

$$m_i U^{n+1}_{H,i} = m_i U^{n+1}_{L,i} + \Delta t^n \sum_{j\neq i} A_{ij}.$$

□

Remark 3.8.4. *The above Lemma implies that the original Zalesak limiter* (3.51) *or the generalized Zalesak limiter* (3.57) *can be applied to* (3.67) *to get a scheme which satisfies the local maximum principle. Numerically, it gives 2nd order accuracy.*

3.9 Numerical tests

All computations are done with $\mathbb{Q}_1$ finite element by using Dealii, a widely used open source library [1] and SSPRK3, see (3.14), is used for time stepping. Eight methods will be considered for all tests:

Method-1. the low-order method with ν^n_K defined in (3.20);

Method-2. the low-order method with ν^n_K defined in (3.17);

Method-3. the high-order method using (3.20);

Method-4. the high-order method using (3.17);

Method-5. the original Zalesak limiter stated in Algorithm 2 using (3.20);

Method-6. the original Zalesak limiter stated in Algorithm 2 using (3.17);

Method-7. the generalized Zalesak limiter stated in Algorithm 3 using (3.20) and ω_{ij} chosen by the method stated in Remark 3.7.5.

Method-8. the generalized Zalesak limiter stated in Algorithm 3 using (3.17) and ω_{ij} chosen by the method stated in Remark 3.7.5.

For comparison, these methods are summarized in Table 3.1.

Table 3.1: Difference between 8 methods

ν_K	Low-order method	High-order method	Zalesak limiter	Generalized Zalesak limiter
(3.20)	Method-1	Method-3	Method-5	Method-7
(3.17)	Method-2	Method-4	Method-6	Method-8

In all computations, the entropy function $E(u)$ in (3.40) is chosen as $E(u) = u^2$.

3.9.1 Transport equation

The linear transport equation is considered first, which is given as following

$$\begin{cases} \partial_t u + \partial_x u(x,y) + \partial_y u(x,y) = 0, \quad \mathbf{x} = (x,y)^\mathsf{T} \in \Omega := [0,1]^2, \\ u(\mathbf{x},0) = \frac{1}{2}[1 - \tanh(\frac{\|\mathbf{x}-\mathbf{x}_0\|^2}{r_0^2} - 1)], \end{cases} \tag{3.77}$$

where $\mathbf{x}_0 = [0.3, 0.3]^\mathsf{T}$ and $r_0 = 0.1$. The final time is chosen as $T = 0.4$. The CFL condition is 0.2. The Dirichlet boundary condition is enforced. The true solution is $u(\mathbf{x}, t) = u(\mathbf{x} - t\mathbf{e}, 0)$ and $\mathbf{e} = (1,1)^\mathsf{T}$ which is obtained by applying the Characteristic method. The error and the convergence are shown in Table 3.2, Table 3.3, Table 3.4 and Table 3.5.

Table 3.2: Transport equation (3.77), Method-1 and Method-2

	Method-1				Method-2			
# dofs	L^2-norm		L^1-norm		L^2-norm		L^1-norm	
289	1.15496e-01	-	4.11962e-02	-	1.22084e-01	-	4.29942e-02	-
1089	9.86404e-02	0.23	3.57306e-02	0.21	1.09131e-01	0.16	3.99346e-02	0.11
4225	7.59799e-02	0.38	2.69216e-02	0.41	8.96276e-02	0.28	3.24611e-02	0.30
16641	5.22835e-02	0.54	1.79035e-02	0.59	6.58497e-02	0.44	2.30315e-02	0.50
66049	3.28825e-02	0.67	1.09275e-02	0.71	4.34975e-02	0.60	1.47040e-02	0.65

Table 3.3: Transport equation (3.77), Method-3 and Method-4

	Method-3				Method-4			
# dofs	L^2-norm		L^1-norm		L^2-norm		L^1-norm	
289	9.28591e-02	-	4.65969e-02	-	9.28591e-02	-	4.65969e-02	-
1089	4.94932e-02	0.91	2.21228e-02	1.07	4.94932e-02	0.91	2.21228e-02	1.07
4225	2.13193e-02	1.22	7.47179e-03	1.57	2.13193e-02	1.22	7.47179e-03	1.57
16641	7.17118e-03	1.57	2.02426e-03	1.88	7.17118e-03	1.57	2.02426e-03	1.88
66049	1.94405e-03	1.88	5.04099e-04	2.01	1.94405e-03	1.88	5.04099e-04	2.01

Table 3.4: Transport equation (3.77), Method-5 and Method-6

	Method-5				Method-6			
# dofs	L^2-norm		L^1-norm		L^2-norm		L^1-norm	
289	8.40235e-02	-	2.84219e-02	-	9.43464e-02	-	3.25561e-02	-
1089	2.95511e-02	1.51	8.69761e-03	1.71	3.77796e-02	1.32	1.10671e-02	1.56
4225	5.56789e-03	2.41	1.58500e-03	2.46	5.88119e-03	2.68	1.56831e-03	2.82
16641	9.58898e-04	2.54	2.37286e-04	2.74	1.02798e-03	2.52	2.45671e-04	2.67
66049	1.56188e-04	2.62	3.88755e-05	2.61	1.72803e-04	2.57	4.05962e-05	2.60

Table 3.5: Transport equation (3.77), Method-7 and Method-8

	Method-7				Method-8			
# dofs	L^2-norm		L^1-norm		L^2-norm		L^1-norm	
289	8.42580e-02	-	2.86157e-02	-	9.52201e-02	-	3.29245e-02	-
1089	3.04748e-02	1.47	9.04557e-03	1.66	3.85208e-02	1.31	1.13067e-02	1.54
4225	5.59227e-03	2.45	1.59093e-03	2.51	5.95133e-03	2.69	1.57076e-03	2.85
16641	9.84874e-04	2.51	2.40697e-04	2.72	1.04576e-03	2.51	2.47854e-04	2.66
66049	1.63540e-04	2.59	3.96433e-05	2.60	1.75671e-04	2.57	4.08508e-05	2.60

The solutions at T are shown in Figure 3.1, Figure 3.2, Figure 3.3 and Figure 3.4, including 25 contour levels with equidistributed values in [0,1]. Note that for both Method-5 and Method-6, the maximum principle is violated, and the bound of $u(T)$ is [-2.03E-5,0.88]. The tiny oscillations around minimum and maximum values can be seen from the

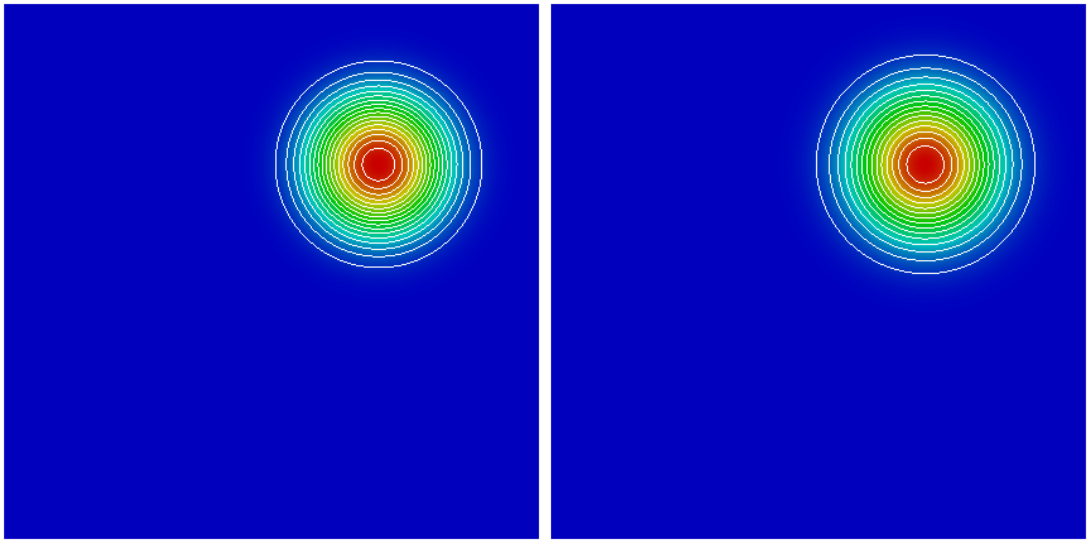


Figure 3.1: Transport equation (3.77), Method-1 and Method-2

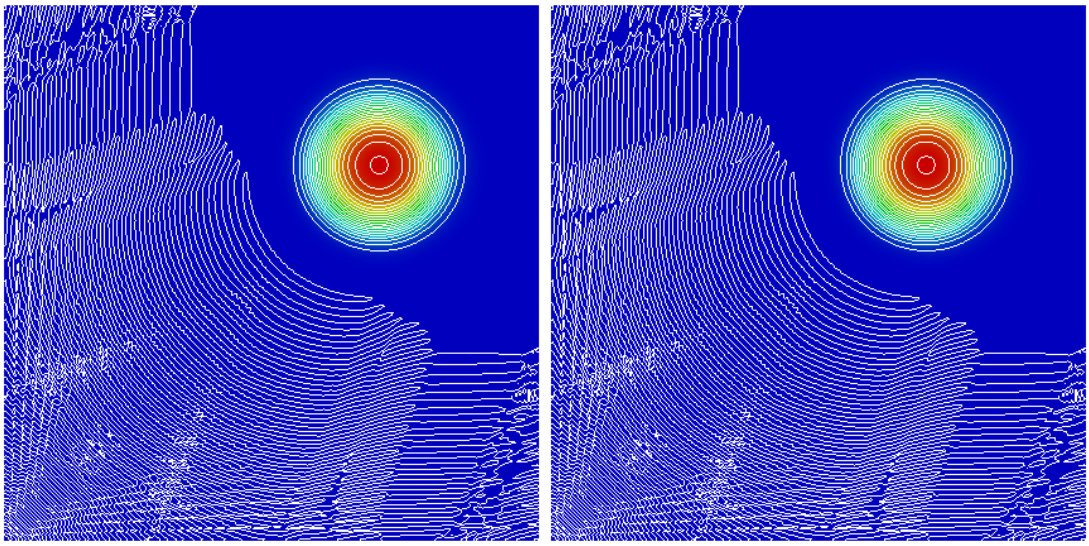


Figure 3.2: Transport equation (3.77), Method-3 and Method-4

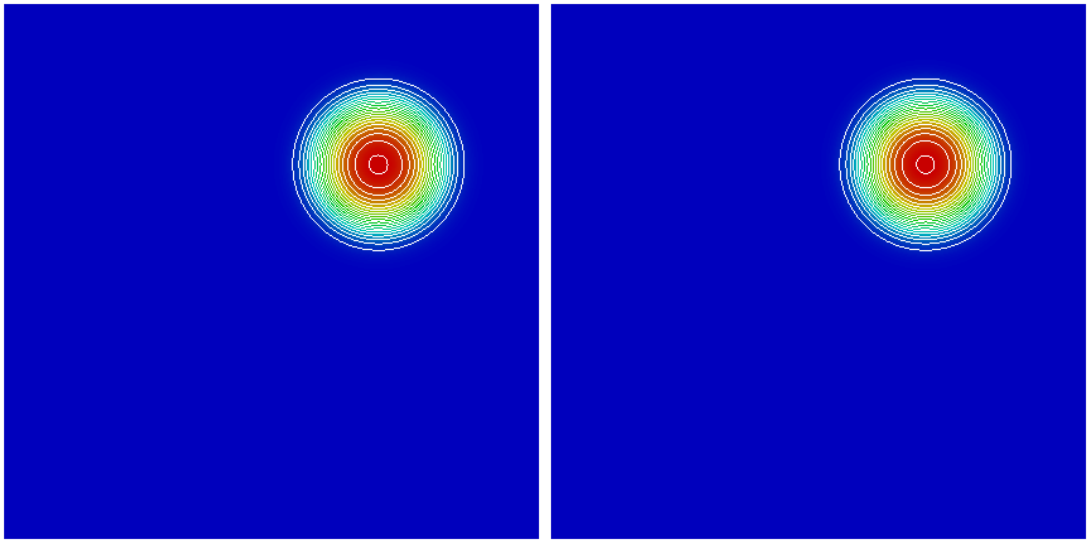


Figure 3.3: Transport equation (3.77), Method-5 and Method-6

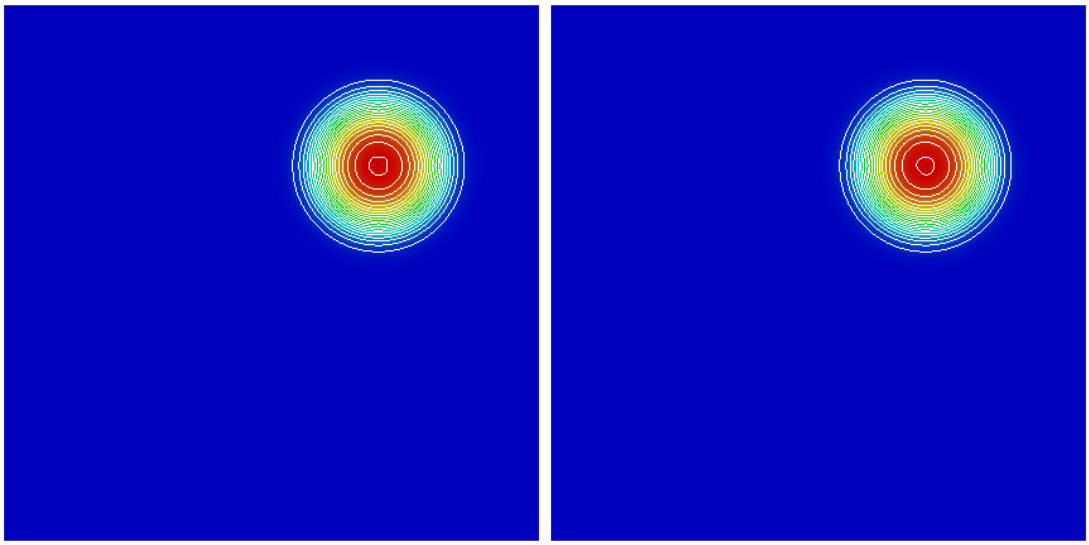


Figure 3.4: Transport equation (3.77), Method-7 and Method-8

Figure 3.3. In this test, we choose $c_E = 0.25$. One can see that the Method-7 and Method-8 has high order accuracy and are maximum principle preserving.

3.9.2 Burgers equation

The Burgers equation is an typical model of nonlinear conservation laws where the flux function is $\mathbf{f}(u) = (\frac{u^2}{2}, \frac{u^2}{2})^\mathsf{T}$. The domain is chosen to be $\Omega = [0,1]^2$. The initial data is chosen as

$$u(\mathbf{x}, 0) = -0.2\mathbb{1}_{x<0.5,y>0.5} - \mathbb{1}_{x>0.5,y>0.5} + 0.5\mathbb{1}_{x<0.5,y<0.5} + 0.8\mathbb{1}_{x>0.5,y<0.5}. \tag{3.78}$$

The true solution to the above special initial data is given as (see, e.g., [40])

$$u(\mathbf{x}, t) = \begin{cases} 0.7\mathbb{1}_{y<0.5+3t/20} - 0.2, & \text{if } x < 0.5 - 0.6t \\ 1.5\mathbb{1}_{y<-8x/7+15/14-15t/28} - 1, & \text{if } 0.5 - 0.6t \le x < 0.5 - 0.25t \\ 1.5\mathbb{1}_{y<x/6+5/12-5t/24} - 1, & \text{if } 0.5 - 0.25t \le x < 0.5 + 0.5t \\ [(2x-1)/(2t)+1]\mathbb{1}_{y<x-5(x+t-0.5)^2/(18t)} - 1, & \text{if } 0.5 + 0.5t \le x < 0.5 + 0.8t \\ 1.8\mathbb{1}_{y<0.5-0.1t} - 1, & \text{if } 0.5 + 0.8t \le x \end{cases} \tag{3.79}$$

The final time is $T = 0.5$ and the CFL condition is 0.2 in all computations of this problem. Eight methods are tested. The error and convergence are shown in Table 3.6, Table 3.7, Table 3.8, and Table 3.9.

Table 3.6: Burgers equation (3.78), Method-1 and Method-2

	Method-1				Method-2			
# dofs	L^2-norm		L^1-norm		L^2-norm		L^1-norm	
81	2.64897e-01	-	1.43882e-01	-	2.44937e-01	-	1.27858e-01	-
289	2.16402e-01	0.29	9.49321e-02	0.60	2.14064e-01	0.19	9.45442e-02	0.44
1089	1.70166e-01	0.35	5.79445e-02	0.71	1.75013e-01	0.29	6.12555e-02	0.63
4225	1.26688e-01	0.43	3.28475e-02	0.82	1.31581e-01	0.41	3.57879e-02	0.78
16641	9.28015e-02	0.45	1.80419e-02	0.86	9.82288e-02	0.42	2.04206e-02	0.81
66049	6.82616e-02	0.44	9.75552e-03	0.89	7.31766e-02	0.42	1.12986e-02	0.85

Table 3.7: Burgers equation (3.78), Method-3 and Method-4

	Method-3				Method-4			
# dofs	L^2-norm		L^1-norm		L^2-norm		L^1-norm	
81	2.48157e-01	-	1.29143e-01	-	2.39089e-01	-	1.19795e-01	-
289	1.86823e-01	0.41	7.59230e-02	0.77	1.84453e-01	0.37	7.38181e-02	0.70
1089	1.34727e-01	0.47	4.32881e-02	0.81	1.33180e-01	0.47	4.24557e-02	0.80
4225	9.90048e-02	0.44	2.46160e-02	0.81	9.73505e-02	0.45	2.40963e-02	0.82
16641	7.08435e-02	0.48	1.35109e-02	0.87	6.88695e-02	0.50	1.31728e-02	0.87
66049	5.18583e-02	0.45	7.51473e-03	0.85	5.10254e-02	0.43	7.46459e-03	0.82

Table 3.8: Burgers equation (3.78), Method-5 and Method-6

	Method-5				Method-6			
# dofs	L^2-norm		L^1-norm		L^2-norm		L^1-norm	
81	2.48772e-01	-	1.27031e-01	-	2.38444e-01	-	1.17295e-01	-
289	1.92371e-01	0.37	7.16007e-02	0.83	1.89146e-01	0.33	6.88159e-02	0.77
1089	1.42581e-01	0.43	3.88545e-02	0.88	1.42077e-01	0.41	3.80019e-02	0.86
4225	1.03708e-01	0.46	2.01204e-02	0.95	1.02359e-01	0.47	1.93406e-02	0.97
16641	7.43954e-02	0.48	1.02713e-02	0.97	7.27011e-02	0.49	9.84184e-03	0.97
66049	5.50187e-02	0.44	5.40869e-03	0.93	5.45869e-02	0.41	5.29556e-03	0.89

Table 3.9: Burgers equation (3.78), Method-7 and Method-8

	Method-7				Method-8			
# dofs	L^2-norm		L^1-norm		L^2-norm		L^1-norm	
81	2.47391e-01	-	1.27382e-01	-	2.37830e-01	-	1.17628e-01	-
289	1.92644e-01	0.36	7.22216e-02	0.82	1.88841e-01	0.33	6.89956e-02	0.77
1089	1.43420e-01	0.43	3.94868e-02	0.87	1.42320e-01	0.41	3.82821e-02	0.85
4225	1.04255e-01	0.46	2.04061e-02	0.95	1.02472e-01	0.47	1.94458e-02	0.98
16641	7.48538e-02	0.48	1.04602e-02	0.96	7.28695e-02	0.49	9.92965e-03	0.97
66049	5.53902e-02	0.43	5.49824e-03	0.93	5.46479e-02	0.42	5.32213e-03	0.90

The solution at T are shown in Figure 3.5, Figure 3.6, Figure 3.7, and Figure 3.8, where 25 contour levels are with equidistributed values in [-1,0.8]. In this test, we choose $c_E = 0.25$.

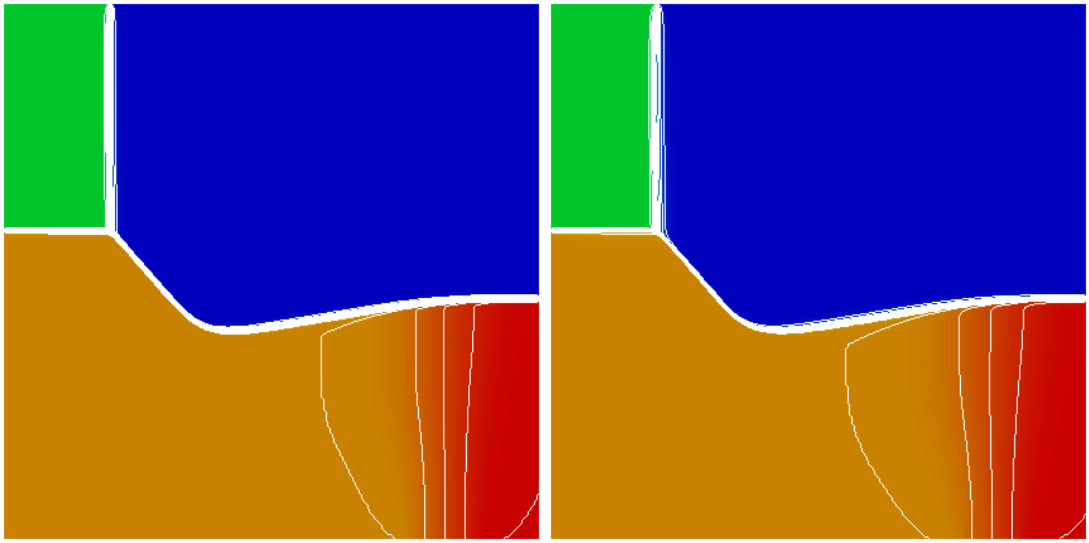


Figure 3.5: Burgers equation (3.78), Method-1 and Method-2

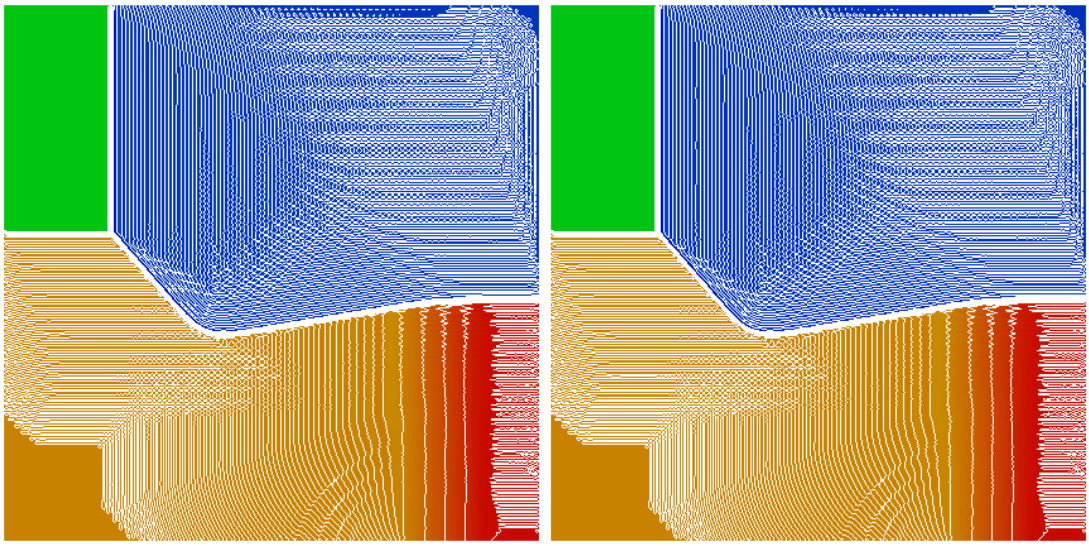


Figure 3.6: Burgers equation (3.78), Method-3 and Method-4

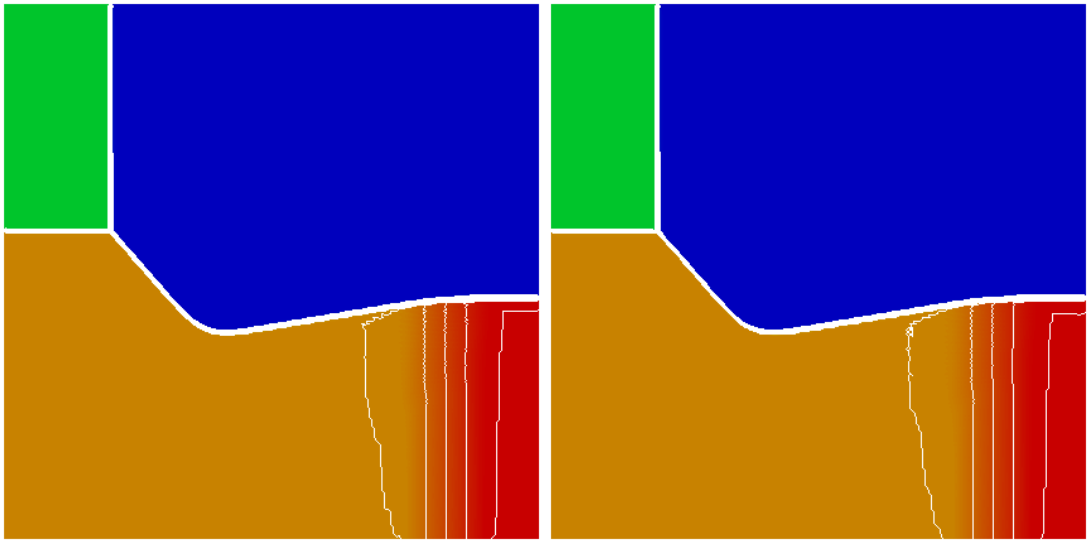


Figure 3.7: Burgers equation (3.78), Method-5 and Method-6

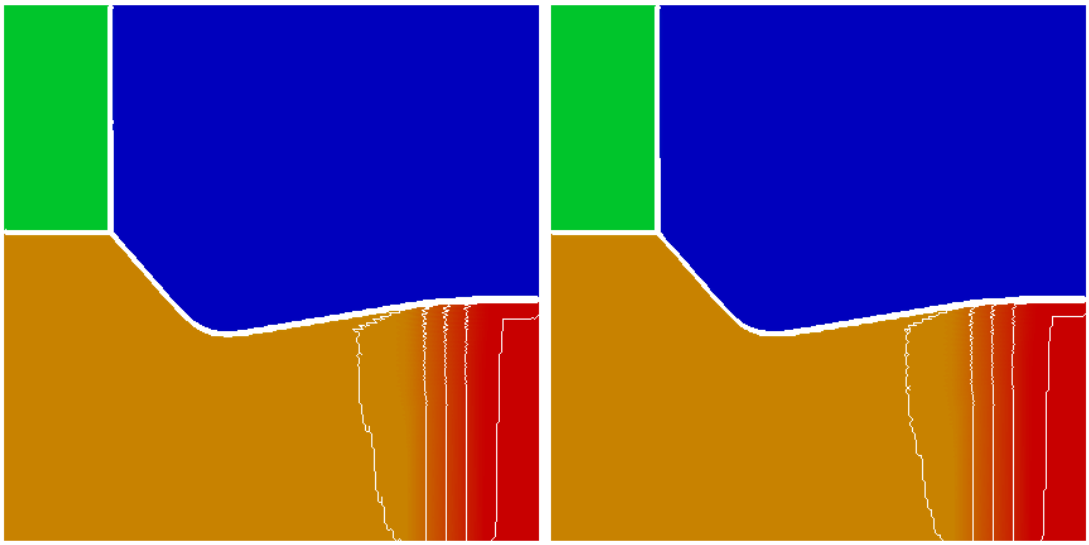


Figure 3.8: Burgers equation (3.78), Method-7 and Method-8

3.9.3 KPP problem

The KPP problem is prosed in [56], where the flux is $\mathbf{f}(u) = (\sin u, \cos u)^{\mathsf{T}}$ and the initial data is given as

$$u(\mathbf{x}, 0) = 3.25\pi \mathbb{1}_{\|\mathbf{x}\|_{\ell^2}<1} + 0.25\pi, \quad \mathbf{x} \in \Omega = [-2, 2]^2. \tag{3.80}$$

The final time is chosen as $T = 1$, which is small enough such that the wave does not reach the boundary $\partial\Omega$. The results of eight methods on the uniform structured mesh with 128×128 cells are shown in Figure 3.9, Figure 3.10, Figure 3.11, and Figure 3.12, where 25 contour levels are with equidistributed values in [0.3,11]. In this test, we choose $c_E = 10$.

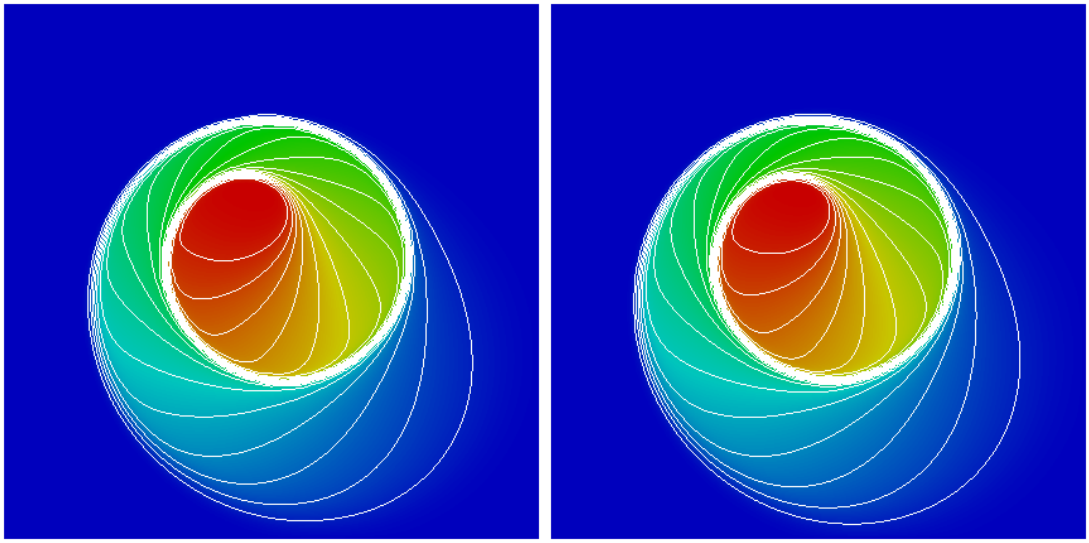


Figure 3.9: KPP equation, Method-1 and Method-2

3.9.4 Buckley–Leverett equation

As for the conservation law with non-convex flux term, another example is the 2D Buckley–Leverett equation (see, e.g., [57], [50, p. 237]). The flux $\mathbf{f}$ in (3.1) is given as $\mathbf{f}(u) := (f(u), g(u))^{\mathsf{T}}$ where $f(u) = \frac{u^2}{u^2+(1-u)^2}$ and $g(u) = f(u)(1 - 5(1 - u)^2)$. The initial

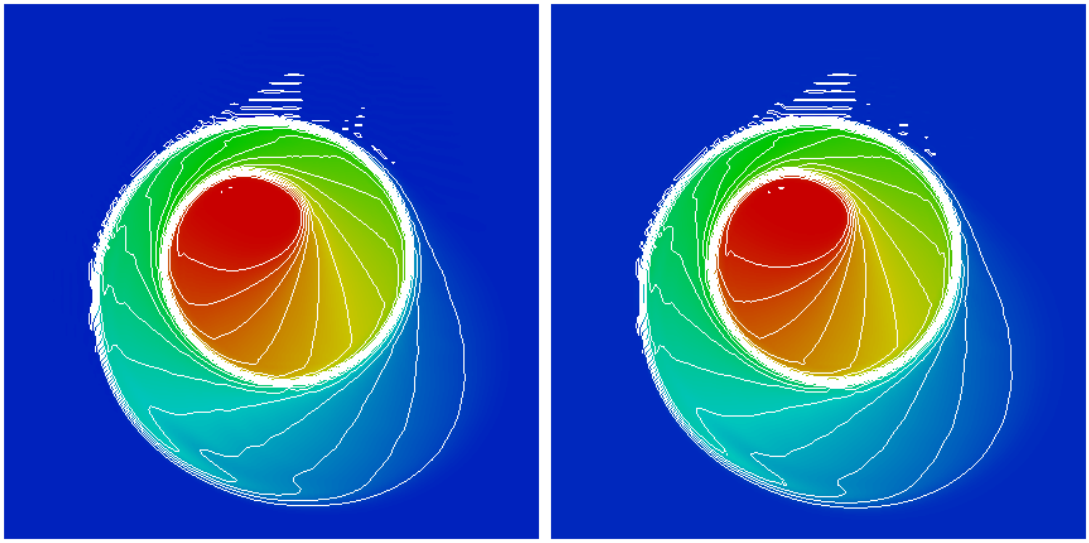


Figure 3.10: KPP equation, Method-3 and Method-4

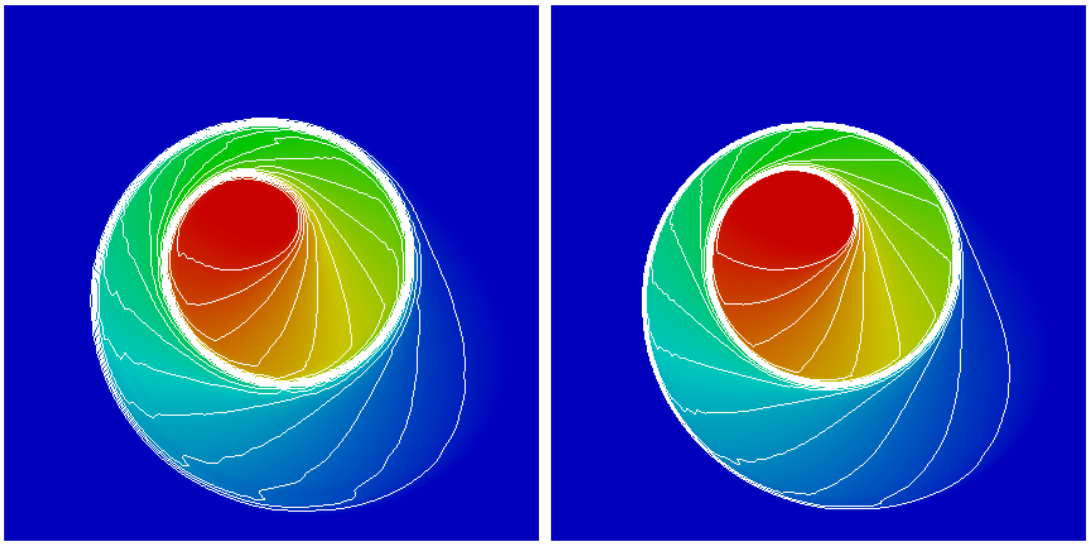


Figure 3.11: KPP equation, Method-5 and Method-6

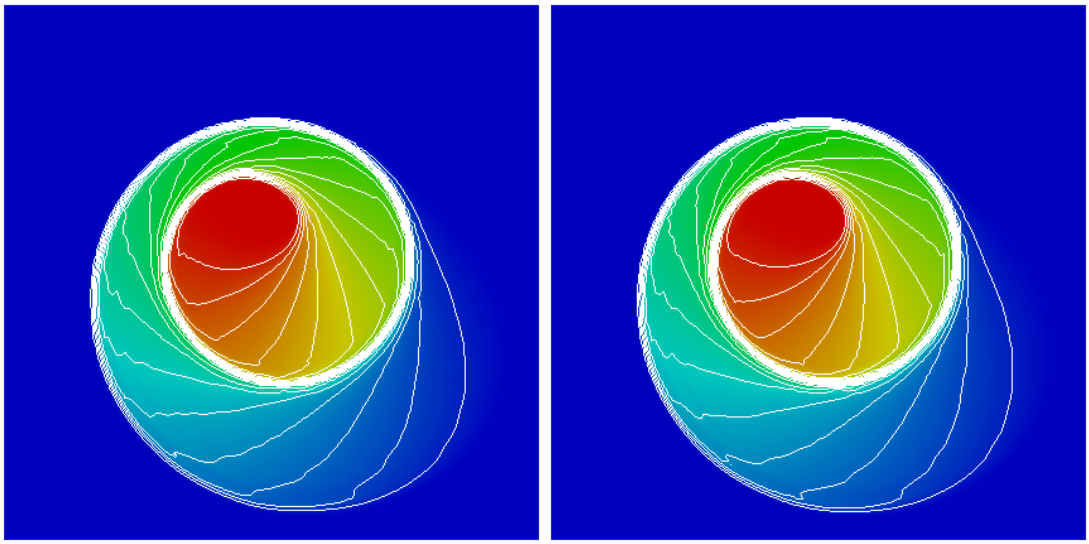


Figure 3.12: KPP equation, Method-7 and Method-8

condition is

$$u(\mathbf{x}, 0) = \mathbb{1}_{\|\mathbf{x}\|<0.5}, \quad \mathbf{x} \in [-1.5, 1.5]^2. \tag{3.81}$$

The final time is chosen as $T = 0.5$. The uniform mesh with 200×200 cells is used in Figure 3.13, Figure 3.14, Figure 3.15, and Figure 3.16 for the comparison to the results in [57], where 25 contour levels are with equidistributed values in [0,1]. In this test, we choose $c_E = 1$.

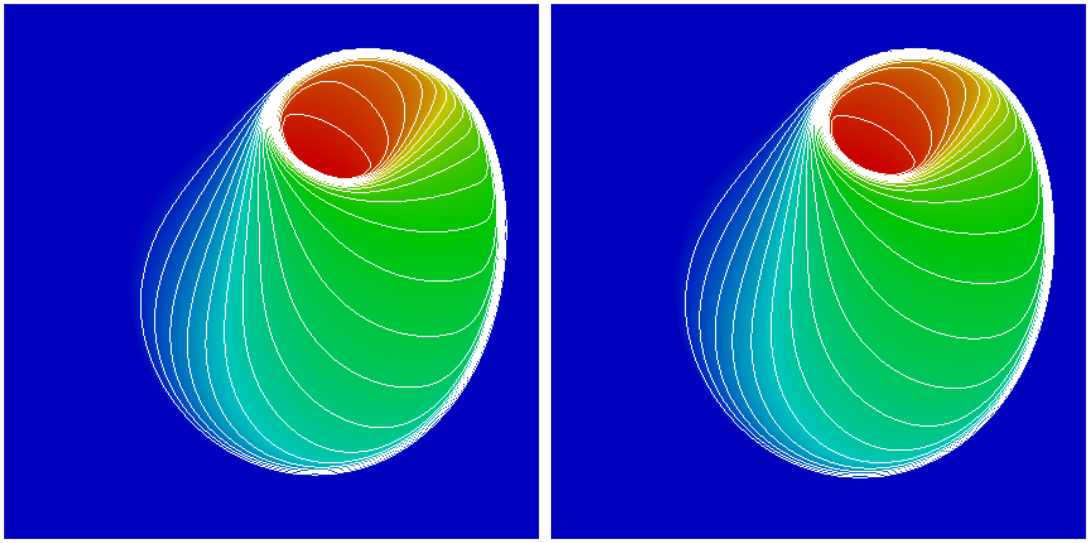


Figure 3.13: Buckley–Leverett equation, Method-1 and Method-2

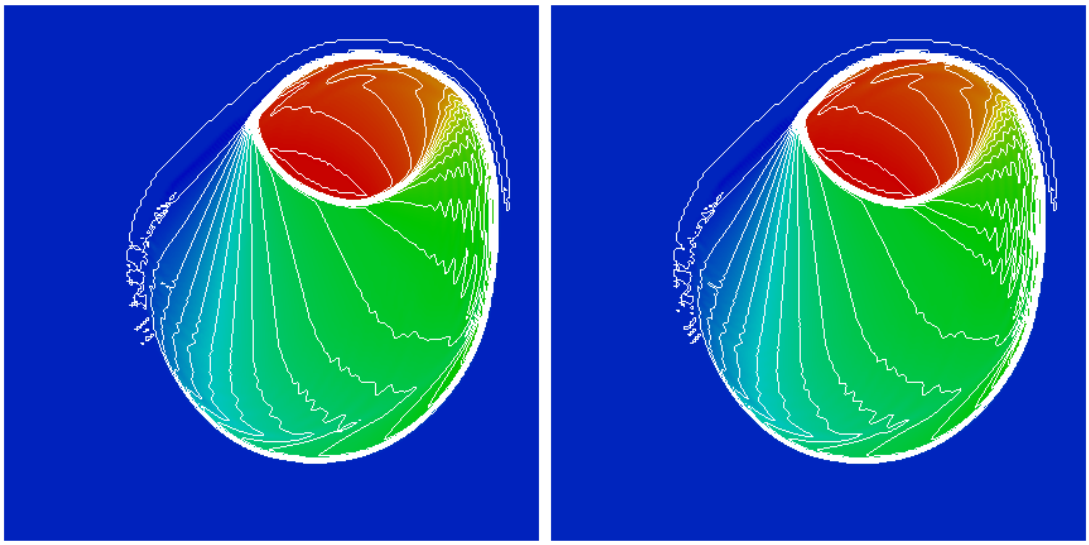


Figure 3.14: Buckley–Leverett equation, Method-3 and Method-4

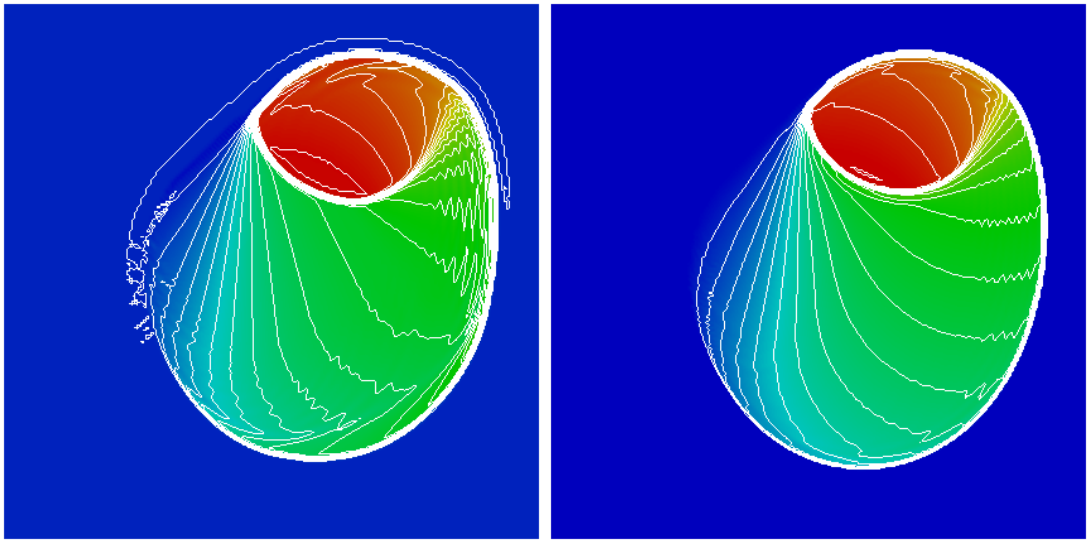


Figure 3.15: Buckley–Leverett Equation, Method-5 and Method-6

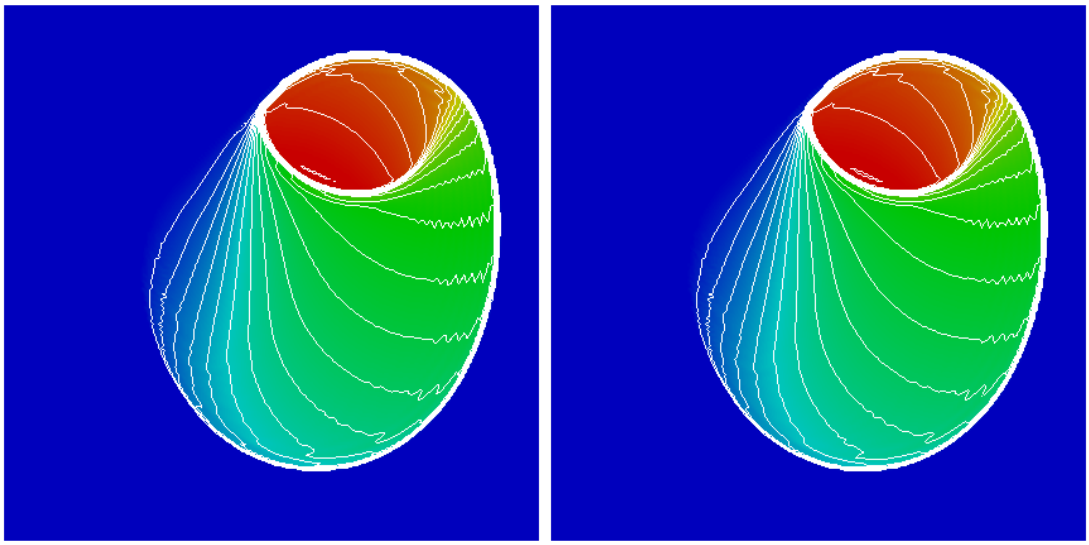


Figure 3.16: Buckley–Leverett equation, Method-7 and Method-8

4. AN ALE METHOD FOR HYPERBOLIC SYSTEMS

Compared to Section 2 and Section 3, in this Section we will use continuous finite element method to solve hyperbolic systems in the Arbitrary Lagrangian Eulerian (ALE) framework and the invariant domain property will be investigated.

4.1 Introduction

Let a hyperbolic system be given as

$$\begin{cases} \partial_t \mathbf{u}(\mathbf{x},t) + \nabla \cdot \mathbf{F}(\mathbf{u}) = 0, \quad \mathbf{x} \in \Omega_t \subset \mathbb{R}^d, \\ \mathbf{u}(\mathbf{x},0) = \mathbf{u}_0(\mathbf{x}), \quad \mathbf{x} \in \Omega_0 \subset \mathbb{R}^d. \end{cases} \tag{4.1}$$

with appropriate initial condition and boundary condition, where $\mathbf{u} \in \mathbb{R}^m$ is the unknown vector and $\mathbf{F} \in \mathbb{R}^{m\times d}$ is called flux.

A convex set $\mathcal{U} \subset \mathbb{R}^k$ is an invariant domain (see, e.g., [14, 48, 49, 25, 6, 43, 44]) for (4.1) if it has the property that $\mathbf{u}(\mathbf{x},t) \in \mathcal{U}$ for all x, t given that $\mathbf{u}(\mathbf{x},0) \in \mathcal{U}$. For example, the maximum principle for a scalar conservation law is equivalent to the property that the interval $[m, M]$ is an invariant domain where m and M are the lower bound and upper bound of the initial data and defined in (3.4).

As a typical prototype of hyperbolic system, the 2D Euler equation can be written as

$$\mathbf{u} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix}, \quad \mathbf{F}(\mathbf{u}) = \begin{pmatrix} \rho u & \rho v \\ \rho u^2 + p & \rho uv \\ \rho vu & \rho v^2 + p \\ u(E+p) & v(E+p) \end{pmatrix}, \tag{4.2}$$

together with an appropriate equation of state, initial condition and boundary conditions, where $E = \rho e + \frac{1}{2}\rho u^2$ is the total energy, and e is the special internal energy. It will be

used in Section 4.7.4 and Section 4.7.6 for numerical study. The set

$$\{\mathbf{u} := (\rho, \rho u, \rho v, E)^{\mathsf{T}} : \rho \geq 0, E - \rho(u^2 + v^2)/2 \geq 0, s \geq r\}$$

for any $r \in \mathbb{R}$ is convex with respect to $\mathbf{u}$ and is a invariant domain of the Euler equations, where s is the special entropy of the system, see [43, (2.15)]. For a general system of hyperbolic conservation laws, a necessary and sufficient condition for a region to be invariant is proposed in [49].

Two continuous finite element methods will be constructed in this Section to preserve the invariant domain property. Since the proposed method only depends on the property of the solution of 1D Riemann problems corresponding to (4.1). We introduce the following assumption about the existence and uniqueness of the solution and the definition about invariant domain property.

Assume that there exists a nonempty admissible set $\mathcal{A}_{\mathbf{F}} \subset \mathbb{R}^m$ such that the following one-dimensional Riemann problem

$$\begin{cases} \partial_t \mathbf{u} + \partial_x(\mathbf{F}(\mathbf{u})\mathbf{n}) = 0, \quad (x,t) \in \mathbb{R}\times\mathbb{R}+, \\ \mathbf{u}(x,0) = \begin{cases} \mathbf{u}_L, & \text{if } x < 0 \\ \mathbf{u}_R, & \text{if } x > 0, \end{cases} \end{cases} \tag{4.3}$$

has a unique entropy solution satisfying solution for any $(\mathbf{u}_L, \mathbf{u}_R) \in \mathcal{A}_{\mathbf{F}}\times\mathcal{A}_{\mathbf{F}}$ and any unit vector $\mathbf{n} \in \mathcal{S}^{d-1}$. We henceforth denote the solution to this problem by $\mathbf{u}(\mathbf{F}, \mathbf{n}, \mathbf{u}_L, \mathbf{u}_R)$. Recall that $(\eta, \mathbf{q})$ is an entropy pair associated with the flux $\mathbf{F}$ if η is convex and

$$\partial_{v_k}(\mathbf{q}(\mathbf{v})\cdot\mathbf{n}) = \sum_{i=1}^{m}\sum_{j=1}^{d} \partial_{v_i}\eta(\mathbf{v})\partial_{v_k}(\mathbf{F}_{ij}(\mathbf{v})n_j), \qquad \forall k \in \{1{:}m\},\ \forall \mathbf{n} \in \mathcal{S}^{d-1}. \tag{4.4}$$

We say that $\mathbf{u}$ is an entropy solution of (4.3) if the following inequality holds

$$\partial_t \eta(\mathbf{u}) + \partial_x(\mathbf{q}(\mathbf{u})\cdot\mathbf{n}) \leq 0. \tag{4.5}$$

in the distribution sense for any entropy pair $(\eta, \mathbf{q})$.

Definition 4.1.1 (Invariant set)**.** *For the problem $\partial_t \mathbf{v} + \nabla \cdot \mathbf{h}(\mathbf{v}) = 0$, we say that a convex set $A \subset \mathcal{A}_{\mathbf{h}} \subset \mathbb{R}^m$ is invariant if for any pair $(\mathbf{v}_L, \mathbf{v}_R) \in A \times A$, any unit vector $\mathbf{n} \in \mathcal{S}^{d-1}$, the average of the entropy solution over $[-\lambda_{\max} t, \lambda_{\max} t]$ remains in A for all $t > 0$, i.e.,*

$$\frac{1}{2t\lambda_{\max}} \int_{-\lambda_{\max} t}^{\lambda_{\max} t} \mathbf{v}(\mathbf{h}, \mathbf{n}, \mathbf{v}_L, \mathbf{v}_R)(x, t)\, \mathrm{d}x \in A,$$

where $\lambda_{\max} = \max\{|\lambda_l|, |\lambda_r|\}$ and λ_l and λ_r satisfy the property that $\mathbf{v}(\mathbf{h}, \mathbf{n}, \mathbf{v}_L, \mathbf{v}_R)(x, t) = \mathbf{v}_L$ for $\frac{x}{t} \le \lambda_l$ and $\mathbf{v}(\mathbf{h}, \mathbf{n}, \mathbf{v}_L, \mathbf{v}_R)(x, t) = \mathbf{v}_R$ for $\frac{x}{t} \ge \lambda_r$.

Remark 4.1.2. *The above definition implies that if A is a convex invariant set, then*

$$\frac{1}{I} \int_I \mathbf{v}(\mathbf{h}, \mathbf{n}, \mathbf{v}_L, \mathbf{v}_R)(x, t)\, \mathrm{d}\mathbf{x} \in A,$$

provided that the interval I satisfies that $(-\lambda_{\max} t, \lambda_{\max} t) \subset I$.

4.2 Weak formulation

For a given function $\widetilde{\mathbf{V}} : \Omega_0 \times \mathbb{R}^+ \to \mathbb{R}^d$, assume the function $\mathbf{\Phi} : \Omega_0 \times \mathbb{R}^+ \to \mathbb{R}^d$

$$\mathbf{x} = \mathbf{\Phi}(\boldsymbol{\xi}, t) = [\Phi_1(\boldsymbol{\xi}, t), \cdots, \Phi_d(\boldsymbol{\xi}, t)]^{\mathsf{T}} \in \mathbb{R}^d, \tag{4.6}$$

satisfies the following equations

$$\begin{cases} \partial_t \mathbf{\Phi}(\boldsymbol{\xi}, t) = \widetilde{\mathbf{V}}(\boldsymbol{\xi}, t), \quad \forall t \in [0, T], \\ \mathbf{\Phi}(\boldsymbol{\xi}, 0) = \boldsymbol{\xi}, \quad \boldsymbol{\xi} \in \Omega_0. \end{cases}$$

and assume its inverse $\mathbf{\Phi}^{-1}(\cdot, t)$ exists for any $t \in [0, T]$) and is smooth enough, at least piecewisely.

In the rest of this Section, we will use the superscript $\tilde{}$ over a letter to emphasize its dependence on the variable $\boldsymbol{\xi}$. In contrast, the corresponding function depending on $\mathbf{x}$

will be denoted by the same letter without using the superscript $\tilde{}$. By this convention, for $t \in [0,T]$, define $\mathbf{V}(\mathbf{x},t) = [V_1(\mathbf{x},t), \cdots, V_d(\mathbf{x},t)]^\mathsf{T} \in \mathbb{R}^d$ as

$$\mathbf{V}(\mathbf{x},t) := \widetilde{\mathbf{V}}(\mathbf{\Phi}^{-1}(\mathbf{x},t),t).$$

It follows that

$$\widetilde{\mathbf{V}}(\boldsymbol{\xi},t) = \mathbf{V}(\mathbf{\Phi}(\boldsymbol{\xi},t),t),$$

and hence $\mathbf{\Phi}$ satisfies

$$\begin{cases} \partial_t \mathbf{\Phi}(\boldsymbol{\xi},t) = \mathbf{V}(\mathbf{\Phi}(\boldsymbol{\xi},t),t), \quad \forall t \in [0,T], \\ \mathbf{\Phi}(\boldsymbol{\xi},0) = \boldsymbol{\xi}, \quad \boldsymbol{\xi} \in \Omega_0. \end{cases} \tag{4.7}$$

Definition 4.2.1. *The Jacobian matrix* $\widetilde{\mathbf{J}} = [\widetilde{\mathbf{J}}_{ij}]_{d\times d} : \Omega_0 \times [0,T] \to \mathbb{R}^{d\times d}$ *is defined by*

$$\widetilde{\mathbf{J}}_{ij}(\boldsymbol{\xi},t) := \frac{\partial \Phi_i}{\partial \xi_j}(\boldsymbol{\xi},t). \tag{4.8}$$

Let $|\widetilde{\mathbf{J}}|$ denote the determinant of the matrix $\widetilde{\mathbf{J}}$.

Lemma 4.2.2. *Assume* $\mathbf{V}(\mathbf{x},t)$ *is smooth enough. For any* $t \in [0,T]$*, the Jacobian matrix* $\widetilde{\mathbf{J}}(\boldsymbol{\xi},t)$ *defined in* (4.2.1) *satisfies the following equality*

$$\partial_t |\widetilde{\mathbf{J}}(\boldsymbol{\xi},t)| = |\widetilde{\mathbf{J}}(\boldsymbol{\xi},t)| \nabla\cdot\mathbf{V}(\mathbf{x},t)|_{\mathbf{x}=\Phi_t(\boldsymbol{\xi})}. \tag{4.9}$$

Proof. By the definition of $\widetilde{\mathbf{J}}$ in (4.2.1), we obtain that

$$
\begin{aligned}
\partial_t|\widetilde{\mathbf{J}}(\boldsymbol{\xi},t)| &= \partial_t|_{\boldsymbol{\xi}}\begin{vmatrix} \frac{\partial\Phi_1}{\partial\xi_1} & \cdots & \frac{\partial\Phi_1}{\partial\xi_d} \\ \vdots & \ddots & \vdots \\ \frac{\partial\Phi_d}{\partial\xi_1} & \cdots & \frac{\partial\Phi_d}{\partial\xi_d} \end{vmatrix} \\
&= \begin{vmatrix} \partial_t\frac{\partial\Phi_1}{\partial\xi_1} & \cdots & \partial_t\frac{\partial\Phi_1}{\partial\xi_d} \\ \vdots & \ddots & \vdots \\ \frac{\partial\Phi_d}{\partial\xi_1} & \cdots & \frac{\partial\Phi_d}{\partial\xi_d} \end{vmatrix} + \cdots + \begin{vmatrix} \frac{\partial\Phi_1}{\partial\xi_1} & \cdots & \frac{\partial\Phi_1}{\partial\xi_d} \\ \vdots & \ddots & \vdots \\ \partial_t\frac{\partial\Phi_d}{\partial\xi_1} & \cdots & \partial_t\frac{\partial\Phi_d}{\partial\xi_d} \end{vmatrix} \\
&= \begin{vmatrix} \frac{\partial}{\partial\xi_1}\partial_t\Phi_1 & \cdots & \frac{\partial}{\partial\xi_d}\partial_t\Phi_1 \\ \vdots & \ddots & \vdots \\ \frac{\partial\Phi_d}{\partial\xi_1} & \cdots & \frac{\partial\Phi_d}{\partial\xi_d} \end{vmatrix} + \cdots + \begin{vmatrix} \frac{\partial\Phi_1}{\partial\xi_1} & \cdots & \frac{\partial\Phi_1}{\partial\xi_d} \\ \vdots & \ddots & \vdots \\ \frac{\partial}{\partial\xi_1}\partial_t\Phi_d & \cdots & \frac{\partial}{\partial\xi_d}\partial_t\Phi_d \end{vmatrix} \\
&= \begin{vmatrix} \frac{\partial}{\partial\xi_1}V_1(\Phi(\xi,t),t) & \cdots & \frac{\partial}{\partial\xi_d}V_1(\Phi(\xi,t),t) \\ \vdots & \ddots & \vdots \\ \frac{\partial\Phi_d}{\partial\xi_1} & \cdots & \frac{\partial\Phi_d}{\partial\xi_d} \end{vmatrix} \\
&\quad + \cdots + \begin{vmatrix} \frac{\partial\Phi_1}{\partial\xi_1} & \cdots & \frac{\partial\Phi_1}{\partial\xi_d} \\ \vdots & \ddots & \vdots \\ \frac{\partial}{\partial\xi_1}V_d(\Phi(\xi,t),t) & \cdots & \frac{\partial}{\partial\xi_d}V_d(\Phi(\xi,t),t) \end{vmatrix} \\
&= \frac{\partial}{\partial x_1}V_1(\Phi(\xi,t),t)\begin{vmatrix} \frac{\partial\Phi_1}{\partial\xi_1} & \cdots & \frac{\partial\Phi_1}{\partial\xi_d} \\ \vdots & \ddots & \vdots \\ \frac{\partial\Phi_d}{\partial\xi_1} & \cdots & \frac{\partial\Phi_d}{\partial\xi_d} \end{vmatrix} + \cdots + \frac{\partial}{\partial x_d}V_d(\Phi(\xi,t),t)\begin{vmatrix} \frac{\partial\Phi_1}{\partial\xi_1} & \cdots & \frac{\partial\Phi_1}{\partial\xi_d} \\ \vdots & \ddots & \vdots \\ \frac{\partial\Phi_d}{\partial\xi_1} & \cdots & \frac{\partial\Phi_d}{\partial\xi_d} \end{vmatrix} \\
&= |\widetilde{\mathbf{J}}(\boldsymbol{\xi},t)|\nabla\cdot\mathbf{V}(\mathbf{x},t)|_{\mathbf{x}=\boldsymbol{\Phi}_t(\boldsymbol{\xi})},
\end{aligned}
$$

where the relation (4.7) is used in the 4th equality. □

For any function $\tilde{\phi}(\boldsymbol{\xi})$, we introduce the function $\phi(\mathbf{x},t)$ defined by

$$\phi(\mathbf{x},t) = \widetilde{\phi}(\boldsymbol{\xi})|_{\mathbf{x}=\Phi(\boldsymbol{\xi},t)}. \tag{4.10}$$

Note that $\tilde{\phi}$ is independent of t. Multiplying (4.1) by the test function $\phi(\mathbf{x},t)$ gives the following weak formulation.

Lemma 4.2.3. *The solution* $\mathbf{u}$ *of (4.1) satisfies the following weak formulation*

$$\frac{d}{dt}\int_{\Omega_t}\mathbf{u}(\mathbf{x},t)\phi(\mathbf{x},t)\,\mathrm{d}\mathbf{x}+\int_{\Omega_t}\phi(\mathbf{x},t)\nabla\cdot[\mathbf{F}(\mathbf{u})-\mathbf{u}\otimes\mathbf{V}]\,\mathrm{d}\mathbf{x}=0, \tag{4.11}$$

for any $\phi(\mathbf{x},t)$ *defined in* (4.10).

Proof.

$$\begin{aligned}
\frac{d}{dt}\int_{\Omega_t}\mathbf{u}(\mathbf{x},t)\phi(\mathbf{x},t)\,\mathrm{d}\mathbf{x} =& \frac{d}{dt}\int_{\Omega_0}\widetilde{\mathbf{u}}(\boldsymbol{\xi},t)\phi(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi}\\
=&\int_{\Omega_0}\phi(\boldsymbol{\xi})\frac{\partial}{\partial t}[\widetilde{\mathbf{u}}(\boldsymbol{\xi},t)|\widetilde{\mathbf{J}}(\boldsymbol{\xi},t)|]\,\mathrm{d}\boldsymbol{\xi}\\
=&\int_{\Omega_0}\phi(\boldsymbol{\xi})[\frac{\partial}{\partial t}|_{\boldsymbol{\xi}}\widetilde{\mathbf{u}}(\boldsymbol{\xi},t)|\widetilde{\mathbf{J}}(\boldsymbol{\xi},t)|+\widetilde{\mathbf{u}}(\boldsymbol{\xi},t)|\frac{\partial}{\partial t}|\widetilde{\mathbf{J}}(\boldsymbol{\xi},t)|]\,\mathrm{d}\boldsymbol{\xi}\\
=&\int_{\Omega_0}\phi(\boldsymbol{\xi})[\nabla_{\mathbf{x}}\mathbf{u}(\mathbf{x},t)\frac{\partial\mathbf{x}}{\partial t}+\partial_t\mathbf{u}(\mathbf{x},t)]|_{\mathbf{x}=\Phi(\boldsymbol{\xi},t)}|\widetilde{\mathbf{J}}(\boldsymbol{\xi},t)|\\
&+\widetilde{\mathbf{u}}(\boldsymbol{\xi},t)\nabla\cdot\mathbf{V}(\mathbf{x},t)|_{\mathbf{x}=\Phi(\boldsymbol{\xi},t)}|\widetilde{\mathbf{J}}(\boldsymbol{\xi},t)|]\,\mathrm{d}\boldsymbol{\xi}\\
=&\int_{\Omega_t}\phi(\mathbf{x},t)[\nabla_{\mathbf{x}}\mathbf{u}(\mathbf{x},t)\mathbf{V}(\mathbf{x},t)+\partial_t\mathbf{u}(\mathbf{x},t)+\mathbf{u}(\mathbf{x},t)\nabla\cdot\mathbf{V}(\mathbf{x},t)]\,\mathrm{d}\mathbf{x}\\
=&\int_{\Omega_t}\phi(\mathbf{x},t)[\nabla_{\mathbf{x}}\mathbf{u}(\mathbf{x},t)\mathbf{V}(\mathbf{x},t)-\nabla\cdot\mathbf{F}(\mathbf{u})+\mathbf{u}(\mathbf{x},t)\nabla\cdot\mathbf{V}(\mathbf{x},t)]\,\mathrm{d}\mathbf{x}\\
=&\int_{\Omega_t}\phi(\mathbf{x},t)[-\nabla\cdot\mathbf{F}(\mathbf{u})+\nabla\cdot(\mathbf{u}\otimes\mathbf{V})(\mathbf{x},t)]\,\mathrm{d}\mathbf{x},
\end{aligned}$$

where we used the property (4.2.2) in the 4th equality and the property

$$\nabla\mathbf{u}(\mathbf{x},t)\mathbf{V}(\mathbf{x},t)+\mathbf{u}(\mathbf{x},t)\nabla\cdot\mathbf{V}(\mathbf{x},t)=\nabla\cdot(\mathbf{u}\otimes\mathbf{V})(\mathbf{x},t) \tag{4.12}$$

in the last equality. □

Remark 4.2.4. *The weak formulation* (4.11) *is in conservative form. In contrast, using*

Lemma 4.2.3, the non-conservative form is given as

$$\int_{\Omega_t} [\frac{d}{dt}|_{\boldsymbol{\xi}} \mathbf{u}(\mathbf{x},t)]\phi(\mathbf{x},t)\,\mathrm{d}\mathbf{x} + \int_{\Omega_t} \phi(\mathbf{x},t)\nabla\cdot\mathbf{F}(\mathbf{u})\,\mathrm{d}\mathbf{x} - \int_{\Omega_t} \nabla\mathbf{u}(\mathbf{x},t)\mathbf{V}(\mathbf{x},t)\,\mathrm{d}\mathbf{x} = 0. \quad (4.13)$$

4.3 Cell mapping

Let $(\mathcal{T}_h^0)_{h>0}$ be a shape-regular sequence of affine matching meshes at initial time and $\mathcal{T}_h^n$ is the mesh at time t^n deformed from $\mathcal{T}_h^0$. Since each cell K^n in $\mathcal{T}_h^n$ is closed we have $\overline{\Omega}_{t^n} = \cup\{K^n : K^n \in \mathcal{T}_h^n\}$. By abuse of notation, we also use K to denote a cell in $\mathcal{T}_h^n$ since the topology of $\mathcal{T}_h^n$ is the same as the topology of $\mathcal{T}_h^0$ although each vertex is moving. Let the geometric reference element be $\{(\widehat{K}, \widehat{P}^{\mathrm{geo}}, \widehat{\Sigma}^{\mathrm{geo}})\}$ (see, e.g., [21, Definition 1.50]) and the map from the reference element to the cell $K \in \mathcal{T}_h^n$ be $T_K^n : \widehat{K} \longrightarrow K$ defined by

$$T_K^n(\widehat{\mathbf{x}}) = \sum_{i\in\{1:n_{\mathrm{sh}}^{\mathrm{geo}}\}} \mathbf{a}^n_{\mathsf{j}^{\mathrm{geo}}(i,K)} \widehat{\theta}_i^{\mathrm{geo}}(\widehat{\mathbf{x}}), \quad (4.14)$$

where the Lagrange nodes of the mesh $\mathcal{T}_h^n$ are $\{\mathbf{a}_j^n : j = 1, \cdots, I^{\mathrm{geo}}\}$. The map from local index to global index is independent of time and denoted by $\mathsf{j}^{\mathrm{geo}} : \{1{:}n_{\mathrm{sh}}^{\mathrm{geo}}\}\times\mathcal{T}_h^n \to \{1{:}I^{\mathrm{geo}}\}$ and $\{\widehat{\theta}_i^{\mathrm{geo}}\}_{i\in\{1:n_{\mathrm{sh}}^{\mathrm{geo}}\}}$ is the Lagrange shape functions associated with the Lagrange nodes $\{\widehat{\mathbf{a}}_i\}_{i\in\{1:n_{\mathrm{sh}}^{\mathrm{geo}}\}}$ of $\widehat{K}$. We introduce the global shape functions in $P^{\mathrm{geo}}(\mathcal{T}_h^n)$ denoted by $\{\psi_i^{\mathrm{geo},n}\}_{i\in\{1:I\}}$

$$\psi^{\mathrm{geo},n}_{\mathsf{j}(i,K)}(\mathbf{x}) = (\widehat{\theta}_i^{\mathrm{geo}} \circ (T_K^n)^{-1})(\mathbf{x}), \quad \forall i \in \{1{:}n_{\mathrm{sh}}^{\mathrm{geo}}\},\ \forall K \in \mathcal{T}_h^n. \quad (4.15)$$

4.4 Finite element space

Let $\{(\widehat{K}, \widehat{P}, \widehat{\Sigma})\}$ be the reference finite element. Assume the basis of $\widehat{P}$ associated to $\widehat{\Sigma}$ on $\widehat{K}$ is denoted by $\{\widehat{\theta}_i\}_{i\in\{1:n_{\mathrm{sh}}\}}$ and satisfies the following property:

$$\widehat{\theta}_i(\mathbf{x}) \geq 0, \quad \sum_{i\in\{1:n_{\mathrm{sh}}\}} \widehat{\theta}_i(\widehat{\mathbf{x}}) = 1, \quad \forall \widehat{\mathbf{x}} \in \widehat{K}. \quad (4.16)$$

The global shape functions in $P(\mathcal{T}_h^n)$ are denoted by $\{\psi_i^n\}_{i\in\{1:I\}}$

$$\psi_{\mathrm{j}(i,K)}^n(\mathbf{x}) = (\widehat{\theta}_i \circ (T_K^n)^{-1})(\mathbf{x}), \quad \forall i \in \{1{:}n_{\mathrm{sh}}\},\ \forall K \in \mathcal{T}_h^n. \tag{4.17}$$

and the following relation holds

$$\psi_i^n(\mathbf{x}) \geq 0, \quad \sum_{i\in\{1:I\}} \psi_i^n(\mathbf{x}) = 1, \quad \forall \mathbf{x} \in \mathbb{R}^d \tag{4.18}$$

owing to the local property (4.16).

We define the scalar-valued function space on the mesh $\mathcal{T}_h^n$ as

$$P(\mathcal{T}_h^n) := \{v \in \mathcal{C}^0(\Omega_{t^n};\mathbb{R}) : v_{|K} \circ T_K^n \in \widehat{P},\ \forall K \in \mathcal{T}_h^n\}, \tag{4.19}$$

where the map T_K^n is defined in (4.14). For any positive integer l, define the vector-valued function space as

$$\mathbf{P}_l(\mathcal{T}_h^n) := [P(\mathcal{T}_h^n)]^l. \tag{4.20}$$

Lemma 4.4.1. *Assume $A \subset \mathbb{R}^d$ is convex. If $\mathbf{V}_i \in A$ for any $i \in \mathcal{I}(K^n)$, then $\mathbf{v}_h(\mathbf{x}) := \sum_i \mathbf{V}_i \psi_i^n$ is in A for any $\mathbf{x} \in K^n$.*

Proof. For fixed $\mathbf{x} \in K^n$, the property (4.18) implies $\mathbf{V}(\mathbf{x})$ is a convex combination of $\mathbf{V}_i$ with $i \in \mathcal{I}(K^n)$. Since A is convex, it follows that $\mathbf{V}(\mathbf{x}) \in A$. □

Note that $\widehat{P}$ may be different from $\widehat{P}^{\mathrm{geo}}$. We introduce the following scalar-valued and vector-valued function space

$$P^{\mathrm{geo}}(\mathcal{T}_h^n) = \{v \in \mathcal{C}^0(\Omega_{t^n};\mathbb{R}) : v_{|K} \circ T_K^n \in \widehat{P}^{\mathrm{geo}},\ \forall K \in \mathcal{T}_h^n\}, \tag{4.21}$$

and

$$\mathbf{P}_m^{\mathrm{geo}}(\mathcal{T}_h^n) = [P^{\mathrm{geo}}(\mathcal{T}_h^n)]^m. \tag{4.22}$$

For convenience, define $\mathcal{I}$ for any set E

$$\mathcal{I}(E) = \{i \in \{1{:}I\} : |\operatorname{supp}(\psi_i^n) \cap E| \neq 0\}, \tag{4.23}$$

and

$$S_i^n = \operatorname{supp}(\psi_i^n). \tag{4.24}$$

4.5 Mesh motion

Assume the mesh velocity over $[t^n, t^{n+1}]$ is given as

$$\widehat{V}(\boldsymbol{\xi}, t) = \sum \mathbf{V}_i^n \psi_i^{\mathrm{geo},n}(\boldsymbol{\xi}), \quad \forall \boldsymbol{\xi} \in \Omega_{t^n}. \tag{4.25}$$

Then the deformation is described as

$$\mathbf{x} = \boldsymbol{\Phi}_h^t(\boldsymbol{\xi}) := \boldsymbol{\xi} + t\widehat{V}(\boldsymbol{\xi}, t) = \boldsymbol{\xi} + t \sum \mathbf{V}_i^n \psi_i^{\mathrm{geo},n}(\boldsymbol{\xi}), \quad \forall t \in [t^n, t^{n+1}]. \tag{4.26}$$

For any cell $K^n \in \mathcal{T}_h^n$, using the reference element $\widehat{K}$, it follows that $K^n = T_K^n(\widehat{K})$ and hence

$$\begin{aligned}
\boldsymbol{\Phi}_h^t(K^n) =& \boldsymbol{\Phi}_h^t(T_K^n(\widehat{K})) \\
=& \{\boldsymbol{\Phi}_h^t(T_K^n(\widehat{\mathbf{x}}), t) : \widehat{\mathbf{x}} \in \widehat{K}\} \\
=& \left\{T_K^n(\widehat{\mathbf{x}}) + t \sum V_{\mathrm{j}^{\mathrm{geo}}(i,K^n)}^n \psi_{\mathrm{j}^{\mathrm{geo}}(i,K^n)}^{\mathrm{geo},n}(T_K^n(\widehat{\mathbf{x}})) : \widehat{\mathbf{x}} \in \widehat{K}\right\} \\
=& \left\{\sum \mathbf{a}_{\mathrm{j}^{\mathrm{geo}}(i,K^n)}^n \widehat{\theta}_i^{\mathrm{geo}}(\widehat{\mathbf{x}}) + t \sum V_{\mathrm{j}^{\mathrm{geo}}(i,K^n)}^n \widehat{\theta}_i^{\mathrm{geo}}(\widehat{\mathbf{x}}) : \widehat{\mathbf{x}} \in \widehat{K}\right\} \\
=& \left\{\sum [\mathbf{a}_{\mathrm{j}^{\mathrm{geo}}(i,K^n)}^n + tV_{\mathrm{j}^{\mathrm{geo}}(i,K^n)}^n] \widehat{\theta}_i^{\mathrm{geo}}(\widehat{\mathbf{x}}) : \widehat{\mathbf{x}} \in \widehat{K}\right\},
\end{aligned}$$

where (4.26) is used in the 3rd equality, and both (4.14) and (4.17) are used in the 4th equality. Then we get the following lemma, see Figure 4.1.

Lemma 4.5.1. *If the mesh velocity is*

$$\widehat{V}(\boldsymbol{\xi},t)=\sum \mathbf{V}_i^n\psi_i^{\mathrm{geo},n}(\boldsymbol{\xi})$$

over $[t^n,t^{n+1}]$*, then*

$$\boldsymbol{\Phi}_h^t\circ T_K^n=T_K^t,\quad \forall K\in\mathcal{T}_h^n, \tag{4.27}$$

where T_K^t *is defined by*

$$T_K^t(\widehat{\mathbf{x}}):=\sum_{i\in\{1:n_{\mathrm{sh}}^{\mathrm{geo}}\}}\mathbf{a}_{\mathrm{j}^{\mathrm{geo}}(i,K)}(t)\widehat{\theta}_i^{\mathrm{geo}}(\widehat{\mathbf{x}}), \tag{4.28}$$

and

$$\mathbf{a}_i(t)=\mathbf{a}_i^n+t\mathbf{V}_i^n,\quad \forall i\in\{1{:}I^{geo}\}. \tag{4.29}$$

Remark 4.5.2. *In particular, taking* $t=t^n$ *and* $t=t^{n+1}$*, we obtain that*

$$\begin{aligned}\boldsymbol{\Phi}_h^{t^n}\circ T_K^n=&T_K^n\\ \boldsymbol{\Phi}_h^{t^{n+1}}\circ T_K^n=&T_K^{n+1}\end{aligned}$$

and hence

$$T_K^{t^n}=T_K^n,\quad T_K^{t^{n+1}}=T_K^{n+1}. \tag{4.30}$$

Remark 4.5.3. *Lemma 4.5.1 tells us that if the velocity is given at each Lagrange node at* t^n*, then the deformation of the cell* K *is always a cell for any* $t\in[t^n,t^{n+1}]$*. In the programming, it means saving the motion of each Lagrangian node are enough to keep the record of the motion of the whole mesh. For example, if* $K\in\mathcal{T}_h^n\subset\mathbb{R}^2$ *is a triangle and* $\widehat{V}(\boldsymbol{\xi},t)|_K\in\mathbb{P}_1$*, then* $\boldsymbol{\Phi}_h^t(K)$ *is still a triangle for any* $t\in[t^n,t^{n+1}]$*. Likewise, if* $K\in\mathcal{T}_h^n$ *is a quadrilateral and* $\widehat{V}(\boldsymbol{\xi},t)|_K\in\mathbb{Q}_1$*, then* $\boldsymbol{\Phi}_h^t(K)$ *is still a quadrilateral for any* $t\in[t^n,t^{n+1}]$*.*

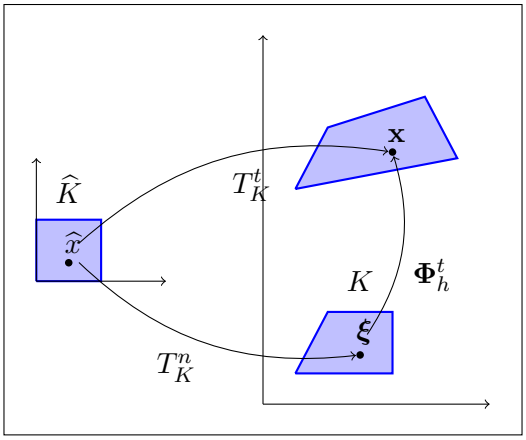


Figure 4.1: Relation between T_K^t and Φ_h^t

Remark 4.5.4. *If the mesh $\mathcal{T}_h^n$ is composed of triangle and quadrilateral, Lemma 4.5.1 still holds since it is just a local property and the only difference is that $\{\widehat{\theta}_i^{\text{geo}}\}$ will differ for different cell types.*

Note that the mesh velocity in Lemma 4.5.1 is a function of $\boldsymbol{\xi}$. In order to use Lemma 4.2.2, we need to find the corresponding function $\mathbf{V}$, a function of $\mathbf{x}$ defined by

$$\mathbf{V}(\mathbf{x}, t) := \widehat{V}((\mathbf{\Phi}_h^t)^{-1}(\mathbf{x}), t). \tag{4.31}$$

From (4.26), it follows that

$$\begin{cases} \frac{\partial}{\partial t}\mathbf{\Phi}_h^t(\boldsymbol{\xi}) = \mathbf{V}(\mathbf{\Phi}_h^t(\boldsymbol{\xi}), t), \quad \forall t \in [t^n, t^{n+1}], \\ \mathbf{\Phi}(\boldsymbol{\xi}, 0) = \boldsymbol{\xi}, \quad \boldsymbol{\xi} \in K, \quad \forall K \in \mathcal{T}_h^n. \end{cases} \tag{4.32}$$

Lemma 4.5.5. *If the mesh velocity is $\widehat{V}(\boldsymbol{\xi}, t) = \sum \mathbf{V}_i^n \psi_i^{\text{geo},n}(\boldsymbol{\xi})$ for $t \in [t^n, t^{n+1}]$, then $\mathbf{V}(\mathbf{x}, t)$ defined in (4.31) satisfies that*

$$\mathbf{V}(\mathbf{x}, t) = \sum_{i \in \{1:\, I^{geo}\}} V_i^n \psi_i^{\text{geo},t}(\mathbf{x}) \in \mathbf{P}_d^{\text{geo}}(\mathcal{T}_h^t), \quad t \in [t^n, t^{n+1}], \tag{4.33}$$

where $\psi_i^{\mathrm{geo},t}(\mathbf{x})$ *is the global shape function in* $\mathbf{P}_d^{\mathrm{geo}}(\mathcal{T}_h^t)$ *locally induced by the map* T_K^t *for each cell* K*, i.e.,*

$$\psi^{\mathrm{geo},t}_{\mathrm{j}^{\mathrm{geo}}(i,K)}(\mathbf{x}) := \widehat{\theta}_i^{\mathrm{geo}} \circ (T_K^t)^{-1}(\mathbf{x}), \tag{4.34}$$

and

$$\mathcal{T}_h^t := \{T_K^t(K) : K \in \mathcal{T}_h^n\}. \tag{4.35}$$

Proof. For fixed $t \in [t^n, t^{n+1}]$ and $K \in \mathcal{T}_h^n$, assume $\mathbf{x} \in \mathbf{\Phi}_h^t(K)$. It follows that

$$\begin{aligned}
\mathbf{V}(\mathbf{x},t) :=& \widehat{V}((\mathbf{\Phi}_h^t)^{-1}(\mathbf{x}),t) \\
=& \sum \mathbf{V}_i^n \psi_i^{\mathrm{geo},n}((\mathbf{\Phi}_h^t)^{-1}(\mathbf{x})) \\
=& \sum V^n_{\mathrm{j}^{\mathrm{geo}}(i,K)} \psi^{\mathrm{geo},n}_{\mathrm{j}^{\mathrm{geo}}(i,K)} \circ (\mathbf{\Phi}_h^t)^{-1}(\mathbf{x}) \\
=& \sum V^n_{\mathrm{j}^{\mathrm{geo}}(i,K)} \widehat{\theta}_i^{\mathrm{geo}} \circ (T_K^n)^{-1} \circ (\mathbf{\Phi}_h^t)^{-1}(\mathbf{x}) \\
=& \sum V^n_{\mathrm{j}^{\mathrm{geo}}(i,K)} \widehat{\theta}_i^{\mathrm{geo}} \circ [\mathbf{\Phi}_h^t \circ T_K^n]^{-1}(\mathbf{x}) \\
=& \sum V^n_{\mathrm{j}^{\mathrm{geo}}(i,K)} \widehat{\theta}_i^{\mathrm{geo}} \circ (T_K^t)^{-1}(\mathbf{x}) \\
=& \sum V^n_{\mathrm{j}^{\mathrm{geo}}(i,K)} \psi^{\mathrm{geo},t}_{\mathrm{j}^{\mathrm{geo}}(i,K)}(\mathbf{x}) \\
=& \sum V_i^n \psi_i^{\mathrm{geo}}(\mathbf{x},t),
\end{aligned}$$

where Lemma 4.5.1 is used in the 6th equality and (4.34) is used in the 7th equality. □

Remark 4.5.6. *Lemma 4.5.5 implies that one can use Lemma 4.2.2. More importantly, since* $\mathbf{V}(\mathbf{x},t) \in \mathbf{P}_d^{\mathrm{geo}}(\mathcal{T}_h^t)$*, its divergence can be computed easily and precisely in any finite element library.*

Let $t \in [t^n, t^{n+1}]$ and in analogy of $\psi_i^{\mathrm{geo}}(\mathbf{x},t)$ defined in (4.34), we introduce

$$\psi^t_{\mathrm{j}(i,K)}(\mathbf{x}) := \widehat{\theta}_i \circ (T_K^t)^{-1}(\mathbf{x}), \quad \forall i \in \{1{:}n_{\mathrm{sh}}\}. \tag{4.36}$$

Corollary 4.5.7.

$$\psi_i^{\mathrm{geo},t}(\mathbf{\Phi}_h^t(\boldsymbol{\xi})) = \psi_i^{\mathrm{geo},n}(\boldsymbol{\xi}), \quad i \in \{1{:}I^{geo}\}, \tag{4.37}$$

and

$$\psi_i^t(\boldsymbol{\Phi}_h^t(\boldsymbol{\xi})) = \psi_i^n(\boldsymbol{\xi}), \quad i \in \{1{:}I\}. \tag{4.38}$$

In particular, (4.30) implies that

$$\psi_i^{\mathrm{geo},n+1}(\boldsymbol{\Phi}_h^{t^{n+1}}(\boldsymbol{\xi})) = \psi_i^{\mathrm{geo},t^{n+1}}(\boldsymbol{\Phi}_h^{t^{n+1}}(\boldsymbol{\xi})) = \psi_i^{\mathrm{geo},n}(\boldsymbol{\xi}), \quad i \in \{1{:}I^{\mathrm{geo}}\}, \tag{4.39}$$

and

$$\psi_i^{n+1}(\boldsymbol{\Phi}_h^{t^{n+1}}(\boldsymbol{\xi})) = \psi_i^{t^{n+1}}(\boldsymbol{\Phi}_h^{t^{n+1}}(\boldsymbol{\xi})) = \psi_i^n(\boldsymbol{\xi}), \quad i \in \{1{:}I\}. \tag{4.40}$$

Definition 4.5.8. *For $t \in [t^n, t^{n+1}]$, the Jacobian $\widetilde{\mathbf{J}}_h^t : \Omega_{t^n} \to \mathbb{R}^{d\times d}$ is defined by*

$$[\widetilde{\mathbf{J}}_h^t]_{ij}(\boldsymbol{\xi}) := \frac{\partial \Phi_i^t}{\partial \xi_j}, \tag{4.41}$$

where Φ_i^t is the i-th term of $\boldsymbol{\Phi}_h^t$ defined in (4.26).

Note that ˜ is used here since $\widetilde{\mathbf{J}}_h^t$ is a function of $\boldsymbol{\xi} \in \Omega_{t^n}$.

Lemma 4.5.9. *Let $(\omega_l, \zeta_l)_{l\in\mathcal{L}}$ be a quadrature such that $\int_0^1 f(\zeta)\,\mathrm{d}\zeta \simeq \sum_{l\in\mathcal{L}} \omega_l f(\zeta_l)$ is exact for all polynomials of degree at most $\max(d-1, 0)$. For fixed $K \in \mathcal{T}_h^n$, $i \in \{1{:}n_{\mathrm{sh}}\}$ and $\tilde{t} \in [t^n, t^{n+1}]$,*

$$\int_{T_K^{\tilde{t}}} \psi_{\mathsf{j}(i,K)}^{\tilde{t}}(\mathbf{x})\,\mathrm{d}\mathbf{x} - \int_{T_K^n} \psi_{\mathsf{j}(i,K)}^n(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi} = \sum_l (\tilde{t}-t^n)\omega_l \int_{T_K^{\bar{t}_l}} \psi_{\mathsf{j}(i,K)}^{\bar{t}_l}(\mathbf{x}) \nabla{\cdot}\mathbf{V}(\mathbf{x}, \bar{t}_l)\,\mathrm{d}\mathbf{x}. \tag{4.42}$$

where $\bar{t}_l$ is defined by

$$\bar{t}_l = t^n + (\tilde{t}-t^n)\zeta_l. \tag{4.43}$$

Proof. Changing variable gives that

$$\int_{T_K^{\tilde{t}}} \psi_{\mathsf{j}(i,K)}^{\tilde{t}}(\mathbf{x})\,\mathrm{d}\mathbf{x} - \int_{T_K^n} \psi_{\mathsf{j}(i,K)}^n(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi} = \int_{T_K^n} \psi_{\mathsf{j}(i,K)}^n(\boldsymbol{\Phi}_h^t(\boldsymbol{\xi}))|\widetilde{\mathbf{J}}_h^{\tilde{t}}|\,\mathrm{d}\boldsymbol{\xi} - \int_{T_K^n} \psi_{\mathsf{j}(i,K)}^n(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi}.$$

By (4.38), it follows that

$$\begin{aligned}\int_{T_K^{\tilde{t}}} \psi_{\mathrm{j}(i,K)}^{\tilde{t}}(\mathbf{x})\,\mathrm{d}\mathbf{x} - \int_{T_K^n} \psi_{\mathrm{j}(i,K)}^{n}(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi} &= \int_{T_K^n} \psi_{\mathrm{j}(i,K)}^{n}(\boldsymbol{\xi})|\widetilde{\mathbf{J}}_h^{\tilde{t}}(\boldsymbol{\xi})|\,\mathrm{d}\boldsymbol{\xi} - \int_{T_K^n} \psi_{\mathrm{j}(i,K)}^{n}(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi} \\ &= \int_{T_K^n} \psi_{\mathrm{j}(i,K)}^{n}(\boldsymbol{\xi})[|\widetilde{\mathbf{J}}_h^{\tilde{t}}(\boldsymbol{\xi})| - 1]\,\mathrm{d}\boldsymbol{\xi} \\ &= \int_{T_K^n} \psi_{\mathrm{j}(i,K)}^{n}(\boldsymbol{\xi}) \left[\int_0^{\tilde{t}-t^n} \partial_s|\widetilde{\mathbf{J}}_h^{s}(\boldsymbol{\xi})|\,\mathrm{d}s\right]\,\mathrm{d}\boldsymbol{\xi}\end{aligned}$$

Since $\boldsymbol{\Phi}_h^t$ defined in (4.26) is linear in t, we see that for fixed $\boldsymbol{\xi}$, $\partial_t|\widetilde{\mathbf{J}}_h^t(\boldsymbol{\xi})|$ is a polynomial in t with degree at most $d-1$. Applying the quadrature trule $(\omega_l, \zeta_l)_{l\in\mathcal{L}}$ yields

$$\int_{T_K^{\tilde{t}}} \psi_{\mathrm{j}(i,K)}(\mathbf{x},t)\,\mathrm{d}\mathbf{x} - \int_{T_K^n} \psi_{\mathrm{j}(i,K)}(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi} = \sum_l (\tilde{t}-t^n)\omega_l \int_{T_K^n} \psi_{\mathrm{j}(i,K)}(\boldsymbol{\xi})\partial_s|\widetilde{\mathbf{J}}_h^{\bar{t}_l}(\boldsymbol{\xi})|\,\mathrm{d}\boldsymbol{\xi},$$

where $\bar{t}_l$ is defined in (4.43). Using Lemma 4.2.2 and the equation (4.32) implies that

$$\partial_s|\widetilde{\mathbf{J}}_h^{\bar{t}_l}|(\boldsymbol{\xi}) = |\widetilde{\mathbf{J}}_h^{\bar{t}_l}(\boldsymbol{\xi})|\nabla\cdot\mathbf{V}(\mathbf{x},\bar{t}_l)|_{\mathbf{x}=\boldsymbol{\Phi}_h^{\bar{t}_l}(\boldsymbol{\xi})}.$$

Therefore, we obtain that

$$\int_{T_K^{\tilde{t}}} \psi_{\mathrm{j}(i,K)}^{\tilde{t}}(\mathbf{x})\,\mathrm{d}\mathbf{x} - \int_{T_K^n} \psi_{\mathrm{j}(i,K)}^{n}(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi} = \sum_l (\tilde{t}-t^n)\omega_l \int_{T_K^{\bar{t}_l}} \psi_{\mathrm{j}(i,K)}^{\bar{t}_l}(\mathbf{x})\nabla\cdot\mathbf{V}(\mathbf{x},\bar{t}_l)\,\mathrm{d}\mathbf{x}.$$

□

Since the topology of the mesh is the same all the time, we obtain the following result.

Corollary 4.5.10. *Let $(\omega_l,\zeta_l)_{l\in\mathcal{L}}$ be a quadrature such that $\int_0^1 f(\zeta)\,\mathrm{d}\zeta \simeq \sum_{l\in\mathcal{L}}\omega_l f(\zeta_l)$ is exact for all polynomials of degree at most $\max(d-1,0)$. For fixed $i\in\{1{:}I\}$ and $\tilde{t}\in[t^n,t^{n+1}]$,*

$$\int_{\Omega_{\tilde{t}}} \psi_{\mathrm{j}(i,K)}^{\tilde{t}}(\mathbf{x})\,\mathrm{d}\mathbf{x} - \int_{\Omega_{t^n}} \psi_{\mathrm{j}(i,K)}^{n}(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi} = \sum_l (\tilde{t}-t^n)\omega_l \int_{\Omega_{\bar{t}_l}} \psi_{\mathrm{j}(i,K)}^{\bar{t}_l}(\mathbf{x})\nabla\cdot\mathbf{V}(\mathbf{x},\bar{t}_l)\,\mathrm{d}\mathbf{x}, \quad (4.44)$$

where $\bar{t}_l$ is defined by

$$\bar{t}_l = t^n + (\tilde{t} - t^n)\zeta_l.$$

Since $\psi_i^{n+1} = \psi_i^{t^{n+1}}$, summing over all $K \in \mathcal{T}_h^n$, we obtain that the following result over the whole domain Ω.

Corollary 4.5.11. *Let $(\omega_l, \zeta_l)_{l\in\mathcal{L}}$ be a quadrature rule such that $\int_0^1 f(\zeta)\,\mathrm{d}\zeta \simeq \sum_{l\in\mathcal{L}} \omega_l f(\zeta_l)$ is exact for all polynomials of degree at most $\max(d-1, 0)$. Then the following equality hold*

$$\int_{\Omega_{t^{n+1}}} \psi_i^{n+1}(\mathbf{x})\,\mathrm{d}\mathbf{x} - \int_{\Omega_{t^n}} \psi_i^n(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi} = \sum_l \Delta t^n \omega_l \int_{\Omega_{t^{n,l}}} \psi_i^{t^{n,l}}(\mathbf{x}) \nabla\cdot\mathbf{V}(\mathbf{x}, t^{n,l})\,\mathrm{d}\mathbf{x}, \tag{4.45}$$

where $t^{n,l}$ is defined by

$$t^{n,l} = t^n + \Delta t^n \zeta_l.$$

When $d = 1$, the quadrature rule $(\omega_l, \zeta_l)_{l\in\mathcal{L}}$ in Corollary 4.5.11 has to be correct for any constant function. In particular, choosing the starting point as the only quadrature point (with weight 1) implies that

Corollary 4.5.12 (1D)**.** *The following equality holds*

$$\int_{\Omega_{t^{n+1}}} \psi_{\mathrm{j}(i,K)}^{n+1}(x, t^{n+1})\,\mathrm{d}x - \int_{\Omega_{t^n}} \psi_{\mathrm{j}(i,K)}^n(\xi)\,\mathrm{d}\xi = \Delta t^n \int_{\Omega_{t^n}} \psi_{\mathrm{j}(i,K)}^n(x, t^n) \nabla\cdot\mathbf{V}(x, t^n)\,\mathrm{d}x,$$

where ψ^{n+1} is related to ψ^n as (4.36)*.*

When $d = 2$, the quadrature rule $(\omega_l, \zeta_l)_{l\in\mathcal{L}}$ in Corollary 4.5.11 has to be correct for any constant function and first order polynomial. In particular, choosing the middle point as the only quadrature point (with weight 1) implies that

Corollary 4.5.13 (2D)**.** *The following equality holds*

$$\int_{\Omega_{t^{n+1}}} \psi^{n+1}_{\mathrm{j}(i,K)}(\mathbf{x}, t^{n+1})\,\mathrm{d}\mathbf{x} - \int_{\Omega_{t^n}} \psi^{n}_{\mathrm{j}(i,K)}(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi} = \Delta t^n \int_{\Omega_{t^{n+1/2}}} \psi^{t^{n+1/2}}_{\mathrm{j}(i,K)}(\mathbf{x})\nabla\cdot\mathbf{V}(\mathbf{x})\,\mathrm{d}\mathbf{x},$$

where ψ^{n+1} *is related to* ψ^n *as* (4.36)*.*

4.6 Finite element method

Assume the function spaces $\widehat{P}^{\mathrm{geo}}$ and $\widehat{P}$ introduced in Section 4.3 and Section 4.4 satisfy the following relation

$$\widehat{P}^{\mathrm{geo}} \subset \widehat{P}. \tag{4.46}$$

It follows that

$$\mathbf{P}_d^{\mathrm{geo}}(\mathcal{T}_h^n) \subset \mathbf{P}_d(\mathcal{T}_h^n), \tag{4.47}$$

and hence there exists a sparse matrix $\mathbb{B}$, independent of n, such that

$$\psi_j^{\mathrm{geo},n} = \sum_i \mathbb{B}_{ij}\psi_i^n, \tag{4.48}$$

and

$$\psi_j^{\mathrm{geo},t} = \sum_i \mathbb{B}_{ij}\psi_i^t. \tag{4.49}$$

From (4.11), we propose a continuous finite element method to solve (4.1). That is to find $\mathbf{u}_h^{n+1} \in \mathbf{P}_m(\mathcal{T}_h^{n+1})$ such that

$$\begin{aligned}(\mathbf{u}_h^{n+1}, \phi(\mathbf{x}, t^{n+1}))_{\Omega_{t^{n+1}},L} - (\mathbf{u}_h^n, \phi(\mathbf{x}, t^n))_{\Omega_{t^n},L} \\ + \Delta t^n \sum_l \omega_l(\nabla\cdot\pi^{n,l}(\mathbf{F}(\mathbf{u}_h^n) - \mathbf{u}_h^n \otimes \mathbf{V}_h^n), \phi(\mathbf{x}, t^{n,l}))_{\Omega_{t^{n,l}}} \\ + B(\mathbf{u}_h^n, \phi(\mathbf{x}, t^n)) = 0,\end{aligned} \tag{4.50}$$

for any $\widetilde{\phi}(\boldsymbol{\xi}) \in \mathbf{P}_m(\mathcal{T}_h^n)$, where $\mathbf{u}_h^n \in \mathbf{P}_m(\mathcal{T}_h^n)$ is the given solution at t^n. The subscript L in $(\cdot,\cdot)_{\Omega,L}$ means the mass lumping technique is used to approximate L^2 inner product

$(\cdot,\cdot)_\Omega$, $t^{n,l} := t^n + \Delta t^n \zeta_l$, $\overline{\Omega_{t^{n,l}}} = \cup\{\mathbf{\Phi}_h^{t^{n,l}}(K) : K \in \mathcal{T}_h^n\}$, $\mathbf{\Phi}_h^t$ is defined in (4.26) with

$$\mathbf{V}_h^n(\boldsymbol{\xi}) := \sum_i \mathbf{V}_i^{\mathrm{geo},n} \psi_i^{\mathrm{geo}}(\boldsymbol{\xi}) = \sum_i \mathbf{V}_i^n \psi_i^n(\boldsymbol{\xi}) \in \mathbf{P}_d^{\mathrm{geo}}(\mathcal{T}_h^n) \subset \mathbf{P}_d(\mathcal{T}_h^n) \tag{4.51}$$

which will be discussed in Section 4.6.8, and ϕ is related to $\widetilde{\phi}$ as follows

$$\phi(\mathbf{x}, t) = \widetilde{\phi} \circ [\mathbf{\Phi}_h^t]^{-1}(\mathbf{x}).$$

The bilinear form B in (4.50) is used to introduce numerical viscosity in the same spirit as in Section 3. The approximation $\pi^{n,l} := \pi^{t^{n,l}}$ and π^t is defined by

$$\pi^t(\mathbf{F}(\mathbf{u}_h^n) - \mathbf{u}_h^n \otimes \mathbf{V}_h^n) := \sum_i [\mathbf{F}(\mathbf{U}_i^n) - \mathbf{U}_i^n \otimes \mathbf{V}_i^n] \psi_i^t(\mathbf{x}), \tag{4.52}$$

for any $t \in [t^n, t^{n+1}]$, where $\psi_i^t(\mathbf{x})$ is defined in (4.36). Assume $(\omega_l, \zeta_l)_{l\in\mathcal{L}}$ appeared in (4.50) is a quadrature rule such that $\int_0^1 f(\zeta)\,\mathrm{d}\zeta \simeq \sum_{l\in\mathcal{L}} \omega_l f(\zeta_l)$ is exact for all polynomials of degree at most $\max(d-1, 0)$. Note that $\phi(\mathbf{x}, t^n) = \widetilde{\phi}(\mathbf{x})$ since $\mathbf{\Phi}_h^{t^n} = I$. The reason to use the quadrature rule $(\omega_l, \zeta_l)_{l\in\mathcal{L}}$ in (4.50) to approximate the second term

$$\int_{t^n}^{t^{n+1}} \left[\int_{\Omega_t} \phi(\mathbf{x}, t) \nabla \cdot [\mathbf{F}(\mathbf{u}) - \mathbf{u} \otimes \mathbf{V}]\,\mathrm{d}\mathbf{x} \right] \mathrm{d}t$$

of (4.11) is due to DGCL, see Theorem 4.6.6.

4.6.1 Discrete form

Introduce

$$\psi_i^{n,l} := \psi_i^{t^{n,l}}, \tag{4.53}$$

$$\mathbf{c}_{ij}^{n,l} := \int_{\Omega_{t^{n,l}}} \nabla \psi_j^{n,l}(\mathbf{x}) \psi_i^{n,l}(\mathbf{x})\,\mathrm{d}\mathbf{x}, \tag{4.54}$$

$$\mathbf{c}_{ij}^n := \sum_{l\in\mathcal{L}} \omega_l \mathbf{c}_{ij}^{n,l}, \tag{4.55}$$

and

$$d_{ij} := -B(\psi_j^{n,l}(\mathbf{x}), \psi_i^{n,l}(\mathbf{x})). \tag{4.56}$$

By (4.40), we have that

$$\psi_i^n(\mathbf{x}, t^{n+1}) = \psi_i^n \circ [\mathbf{\Phi}_h^{t^{n+1}}]^{-1}(\mathbf{x}) = \psi_i^{n+1}(\mathbf{x}),$$

which implies that

$$\begin{aligned}(\mathbf{u}_h^{n+1}, \phi_i(\mathbf{x}, t^{n+1}))_{\Omega_{t^{n+1}},L} =& (\mathbf{u}_h^{n+1}, \psi_i^{n+1})_{\Omega_{t^{n+1}},L} \\ =& \sum_j \mathbf{U}_j^{n+1} (\psi_j^{n+1}(\mathbf{x}), \psi_i^{n+1})_{\Omega_{t^{n+1}},L} \\ =& \sum_j \mathbf{U}_j^{n+1} m_j^{n+1},\end{aligned}$$

with

$$m_j^{n+1} := \int_{\Omega_{t^{n+1}}} \psi_j^{n+1}(\mathbf{x})\, \mathrm{d}\mathbf{x}. \tag{4.57}$$

Define

$$\mathbf{F}_j^n := \mathbf{F}(\mathbf{U}_j^n). \tag{4.58}$$

Assume $[d_{ij}]$ satisfies the following three properties (i) $d_{ij}^n = 0$ if $j \notin \mathcal{I}(S_i^n)$; (ii) $d_{ij}^n \geq 0$ and

$$d_{ij}^n = d_{ji}^n \tag{4.59}$$

if $i \neq j$; (iii) $d_{ii} := \sum_{j \neq i} -d_{ji}^n$.

Choosing $\widetilde{\phi}(\boldsymbol{\xi}) = \psi_i(\boldsymbol{\xi})$ in (4.50) implies the discrete form

$$\begin{aligned}&m_i^{n+1}\mathbf{U}_i^{n+1} - m_i^n \mathbf{U}_i^{n+1} \\ &\qquad + \Delta t^n \sum_{j \in \mathcal{I}(S_i^n)} [\mathbf{F}_j^n - \mathbf{U}_j^n \otimes \mathbf{V}_j^n] \mathbf{c}_{ij}^n - \sum_{j \in \mathcal{I}(S_i^n)} d_{ij} \mathbf{U}_j^n = 0, \quad \forall i.\end{aligned} \tag{4.60}$$

4.6.2 Conservation

Theorem 4.6.1. *The scheme* (4.50) *or* (4.60) *is conservative, i.e.,*

$$\int_{\Omega_{t^n}} \mathbf{u}_h^n \,\mathrm{d}\mathbf{x} = \int_{\Omega_{t^0}} \mathbf{u}_h^n \,\mathrm{d}\mathbf{x}, \quad \forall n. \tag{4.61}$$

Proof. Since

$$\int_{\Omega_{t^{n+1}}} \mathbf{u}_h^{n+1} \,\mathrm{d}\mathbf{x} = \sum_i m_i^{n+1} \mathbf{U}_i^{n+1},$$

and the sum of each column of $\mathbf{c}_{ij}$ and d_{ij} is 0, i.e.,

$$\sum_i \mathbf{c}_{ij} = 0, \quad \sum_i d_{ij} = 0,$$

as a result of the definition of $\mathbf{c}_{ij}$ in (4.55), the assumption on the boundary conditions and the assumption on d_{ij} in (4.59), summing (4.60) over i implies that

$$\int_{\Omega_{t^{n+1}}} \mathbf{u}_h^{n+1} \,\mathrm{d}\mathbf{x} = \int_{\Omega_{t^n}} \mathbf{u}_h^n \,\mathrm{d}\mathbf{x}.$$

Since this is true for all n, it completes the proof. □

Remark 4.6.2. *The form* (4.50)*, as a discretization of the weak formulation* (4.11)*, is called conservation form since Theorem 4.6.1.*

4.6.3 Discrete geometric conservation law

Lemma 4.6.3 (Non-conservative form)**.** *The scheme* (4.50) *or* (4.60) *is equivalent to*

$$m_i^{n+1} \frac{\mathbf{U}_i^{n+1} - \mathbf{U}_i^n}{\Delta t} = \sum_{j \in \mathcal{I}(S_i^n)} ((\mathbf{U}_j^n - \mathbf{U}_i^n) \otimes \mathbf{V}_j^n - \mathbf{F}_j^n)\mathbf{c}_{ij}^n + \sum_{j \in \mathcal{I}(S_i^n)} d_{ij}^n \mathbf{U}_j^n, \quad \forall i. \tag{4.62}$$

Proof. Since

$$m_i^{n+1}\mathbf{U}_i^{n+1} - m_i^n\mathbf{U}_i^{n+1} = m_i^{n+1}[\mathbf{U}_i^{n+1} - \mathbf{U}_i^n] + [m_i^{n+1} - m_i^n]\mathbf{U}_i^n$$

and by Corollary 4.5.11,

$$m_i^{n+1} - m_i^n = \sum_l \Delta t^n \omega_l \int_{\Omega_{t^{n,l}}} \psi_i^{t^{n,l}}(\mathbf{x}) \nabla\cdot\mathbf{V}(\mathbf{x}, t^{n,l})\, \mathrm{d}\mathbf{x}$$

with $t^{n,l}$ defined in (4.50), and by Lemma 4.5.5 and the equality (4.51),

$$\mathbf{V}(\mathbf{x}, t^{n,l}) = \sum_j V_j^{\mathrm{geo},n} \psi_j^{\mathrm{geo},t^{n,l}}(\mathbf{x}) = \sum_j V_j^n \psi_j^{t^{n,l}}(\mathbf{x}) = \sum_j \mathbf{V}_j^n \psi_j^{n,l}(\mathbf{x})$$

we obtain that

$$\nabla\cdot\mathbf{V}(\mathbf{x}, t^{n,l}) = \sum_j \mathbf{V}_j^n \cdot \nabla\psi_j^{n,l}(\mathbf{x})$$

and hence

$$m_i^{n+1} - m_i^n = \Delta t^n \sum_{j\in\mathcal{I}(S_i^n)} \mathbf{V}_j^n \cdot \mathbf{c}_{ij}^n.$$

It implies that

$$[m_i^{n+1} - m_i^n]\mathbf{U}_i^n = \Delta t^n \sum_{j\in\mathcal{I}(S_i^n)} [\mathbf{U}_i^n \otimes \mathbf{V}_j^n]\mathbf{c}_{ij}^n.$$

Therefore, the scheme (4.60) is equivalent to (4.62). □

Remark 4.6.4. *From the proof of the above lemma, we also get an important relation*

about m_i^n *as follows*

$$m_i^{n+1} - m_i^n = \Delta t^n \sum_{j \in \mathcal{I}(S_i^n)} \mathbf{V}_j^n \cdot \mathbf{c}_{ij}^n \tag{4.63}$$

as a result of Corollary 4.5.11, where $\mathbf{c}_{ij}^n$ *is defined in* (4.55).

Definition 4.6.5. *The algorithm is said to satisfy Discrete Geometric Conservation Law (DGCL) (see, e.g., [22, 23, 53, 26] and the references therein) if* $\mathbf{u}_h^{n+1}(\mathbf{x}) = \mathbf{C}$ *under the condition that* $\mathbf{u}_h^n(\mathbf{x}) = \mathbf{C} \in \mathbb{R}^m$.

Theorem 4.6.6 (DGCL)**.** *If* $\mathbf{U}_j^n = \mathbf{C} \in \mathbb{R}^m$ *for all* $j \in \mathcal{I}(S_i^n)$*, then* $\mathbf{U}_i^{n+1} = \mathbf{C}$*. In particular, it implies that the scheme* (4.50) *preserve constant states.*

Proof. From (4.60), since the sum of each row of $\mathbf{c}_{ij}$ and d_{ij} is 0, it follows that

$$\begin{aligned} m_i^{n+1}\mathbf{U}_i^{n+1} =& m_i^n \mathbf{C} - \Delta t^n \sum_{j \in \mathcal{I}(S_i^n)} [\mathbf{F}(\mathbf{C}) - \mathbf{C} \otimes \mathbf{V}_j^n] \mathbf{c}_{ij}^n + \sum_{j \in \mathcal{I}(S_i^n)} d_{ij} \mathbf{C} \\ =& \mathbf{C}[m_i^n + \sum_{j \in \mathcal{I}(S_i^n)} \mathbf{V}_j^n \cdot \mathbf{c}_{ij}^n]. \end{aligned}$$

Therefore, using (4.63) implies that $\mathbf{U}_i^{n+1} = \mathbf{C}$. □

4.6.4 Invariant domain property

For any unit vector $\mathbf{n}$, define $\mathbf{v}(\mathbf{g}, \mathbf{n}, \mathbf{U}_l, \mathbf{U}_r)$ to be the unique entropy solution of the following one-dimensional Riemann problem:

$$\partial_t \mathbf{v} + \partial_x (\mathbf{g}(\mathbf{v}) \cdot \mathbf{n}) = 0, \quad (x,t) \in \mathbb{R} \times \mathbb{R}_+, \quad \mathbf{v}(x,0) = \begin{cases} \mathbf{U}_l & \text{if } x < 0 \\ \mathbf{U}_r & \text{if } x > 0. \end{cases} \tag{4.64}$$

with the largest wave speed $\lambda_{\max}(\mathbf{g}, \mathbf{n}, \mathbf{U}_l, \mathbf{U}_r)$.

Remark 4.6.7. *Since* $\mathbf{v}(\mathbf{g}, \mathbf{n}, \mathbf{U}_l, \mathbf{U}_r)$ *is assumed to be the entropy solution of* (4.64)*, the*

conservation property implies that if $\bar{t}\lambda_{\max}(\mathbf{g},\mathbf{n},\mathbf{U}_l,\mathbf{U}_r) \le \frac{1}{2}$*, then*

$$\int_{-\frac{1}{2}}^{\frac{1}{2}} \mathbf{v}(\mathbf{g},\mathbf{n},\mathbf{U}_l,\mathbf{U}_r)(t,x)\,\mathrm{d}x = \frac{1}{2}[\mathbf{U}_l+\mathbf{U}_r] - \bar{t}[\mathbf{g}(\mathbf{U}_r)\cdot\mathbf{n} - \mathbf{g}(\mathbf{U}_l)\cdot\mathbf{n}]. \tag{4.65}$$

Remark 4.6.8. *If* $(\eta,\mathbf{q})$ *is an entropy pair of* (4.64)*, by entropy inequality* (4.5) *and Jensen's inequality, we obtain that*

$$\eta\left(\int_{-\frac{1}{2}}^{\frac{1}{2}} \mathbf{v}(\mathbf{g},\mathbf{n},\mathbf{U}_l,\mathbf{U}_r)(t,x)\,\mathrm{d}x\right) \le \frac{1}{2}[\eta(\mathbf{U}_l)+\eta(\mathbf{U}_r)] - \bar{t}[\mathbf{q}(\mathbf{U}_r)\cdot\mathbf{n} - \mathbf{q}(\mathbf{U}_l)\cdot\mathbf{n}]. \tag{4.66}$$

For $i \neq j$, we choose d_{ij} as

$$d_{ij}^n = \max(\lambda_{\max}(\mathbf{g}_j^n,\mathbf{n}_{ij}^n,\mathbf{U}_i^n,\mathbf{U}_j^n)\|\mathbf{c}_{ij}^n\|_{\ell^2}, \lambda_{\max}(\mathbf{g}_i^n,\mathbf{n}_{ji}^n,\mathbf{U}_j^n,\mathbf{U}_i^n)\|\mathbf{c}_{ji}^n\|_{\ell^2}) \tag{4.67}$$

where $\mathbf{n}_{ij}^n \in \mathcal{S}^{d-1}$ is the unit direction vector of $\mathbf{c}_{ij}^n$ and

$$\mathbf{g}_j^n(\mathbf{v}) := \mathbf{F}(\mathbf{v}) - \mathbf{v}\otimes\mathbf{V}_j^n. \tag{4.68}$$

Moreover, choose $d_{ii} = -\sum_{j\neq i} d_{ij}$. The assumption (4.59) on d_{ij} is satisfied.

Define

$$h^n := \min_{i,l\in\mathcal{L}} h_i^{n,l}, \quad h_i^{n,l} := \min_{j\in\mathcal{I}(S_i^n)} \frac{1}{\|\|\nabla\psi_j\|_{\ell^2}\|_{L^\infty(S_{ij}^{n,l})}}. \tag{4.69}$$

where $S_{ij}^{n,l} := \mathrm{supp}(\psi_i^{n,l}) \cap \mathrm{supp}(\psi_j^{n,l})$.

Define

$$\kappa^n := \max_i \kappa_i^n, \quad \kappa_i^n := \frac{\sum_{j\in\mathcal{I}(S_i^n)\setminus\{i\},l\in\mathcal{L}} \omega_l \int_{S_{ij}^{n,l}} \psi_i^{n,l}(\mathbf{x})\,\mathrm{d}\mathbf{x}}{\sum_{l\in\mathcal{L}} \omega_l \int_{\Omega^{n,l}} \psi_i^{n,l}(\mathbf{x})\,\mathrm{d}\mathbf{x}}. \tag{4.70}$$

Define

$$\lambda^n := \max_i \lambda_i^n, \quad \lambda_i^n := \max_{j\in\mathcal{I}(S_i^n)} (\lambda_{\max}(\mathbf{g}_j^n,\mathbf{n}_{ij}^n,\mathbf{U}_i^n,\mathbf{U}_j^n), \lambda_{\max}(\mathbf{g}_i^n,\mathbf{n}_{ji}^n,\mathbf{U}_j^n,\mathbf{U}_i^n)). \tag{4.71}$$

Theorem 4.6.9 (Invariant Domain Property)**.** *Let* $A \subset \mathcal{A}_{\mathbf{F}}$ *be an invariant set of* (4.1) *as*

defined in Defition 4.1.1 and d_{ij} is chosen as in (4.67) *with* $\mathbf{c}_{ij}^n$ *defined in* (4.55). *Assume* $m_i^{n+1} > 0$ *for all* i. *If* $\{\mathbf{U}_j^n : j \in \mathcal{I}(S_i^n)\} \subset A$ *and* Δt^n *satisfies the following CFL condition*

$$2\Delta t^n \frac{\kappa^n \max_i \lambda_i^n}{h^n} \max_i \frac{\sum_{l\in\mathcal{L}} \omega_l \int_{\Omega^{n,l}} \psi_i^{n,l}(\mathbf{x})\,\mathrm{d}\mathbf{x}}{m_i^{n+1}} \le 1, \tag{4.72}$$

where h^n *and* κ^n *are defined in* (4.69) *and* (4.70), *then the solution of the scheme* (4.60) *has the property that* $\mathbf{U}_i^{n+1} \in A$. *In particular, if* $\{\mathbf{U}_j^n : \forall j\} \subset A$, *then* $\{\mathbf{U}_j^{n+1} : \forall j\} \subset A$.

Proof. By Lemma 4.6.3, the scheme (4.60) is equivalent to

$$m_i^{n+1}\frac{\mathbf{U}_i^{n+1} - \mathbf{U}_i^n}{\Delta t^n} = \sum_{j\in\mathcal{I}(S_i^n)} ((\mathbf{U}_j^n - \mathbf{U}_i^n)\otimes \mathbf{V}_j^n - \mathbf{F}_j^n)\mathbf{c}_{ij}^n + \sum_{j\in\mathcal{I}(S_i^n)} d_{ij}^n \mathbf{U}_j^n.$$

Since the sum of each row $\sum_j \mathbf{c}_{ij}^n = 0$ for any fixed i, it follows that

$$\sum_j a_j \mathbf{c}_{ij}^n = \sum_{j\neq i} (a_j - a_i)\mathbf{c}_{ij}^n. \tag{4.73}$$

It implies that

$$\sum_{j\neq i} [\mathbf{F}_j^n + (\mathbf{U}_i^n - \mathbf{U}_j^n)\otimes \mathbf{V}_j^n]\mathbf{c}_{ij}^n = \sum_{\substack{j\neq i \\ j\in\mathcal{I}(S_i^n)}} [\mathbf{F}_j^n - \mathbf{F}_i^n + (\mathbf{U}_i^n - \mathbf{U}_j^n)\otimes \mathbf{V}_j^n]\mathbf{c}_{ij}^n.$$

Since the sum of each row of $[d_{ij}]$ is zero, i.e., $\sum_j d_{ij} = 0$, we have that

$$\begin{aligned}\sum_{j\in\mathcal{I}(S_i^n)} d_{ij}\mathbf{U}_j^n &= \sum_{j\in\mathcal{I}(S_i^n)} (\mathbf{U}_j^n + \mathbf{U}_i^n)d_{ij} \\ &= \sum_{\substack{j\neq i \\ j\in\mathcal{I}(S_i^n)}} (\mathbf{U}_j^n + \mathbf{U}_i^n)d_{ij} + 2\mathbf{U}_i^n d_{ii}.\end{aligned}$$

Then we obtain that

$$m_i^{n+1}\mathbf{U}_i^{n+1} = [m_i^{n+1} + 2\Delta t^n d_{ii}]\mathbf{U}_i^n \\ - \Delta t^n \sum_{\substack{j\neq i \\ j\in\mathcal{I}(S_i^n)}} \left\{[\mathbf{F}_j^n - \mathbf{F}_i^n + (\mathbf{U}_i^n - \mathbf{U}_j^n)\otimes \mathbf{V}_j^n]\mathbf{c}_{ij}^n + (\mathbf{U}_j^n + \mathbf{U}_i^n)d_{ij}\right\} \quad (4.74)$$

That is

$$\begin{aligned}\mathbf{U}_i^{n+1} &= [1 + \frac{2\Delta t^n d_{ii}}{m_i^{n+1}}]\mathbf{U}_i^n \\ &\quad - \sum_{\substack{j\neq i \\ j\in\mathcal{I}(S_i^n)}} \frac{2\Delta t^n d_{ij}}{m_i^{n+1}} \left\{[\mathbf{F}_j^n - \mathbf{F}_i^n + (\mathbf{U}_i^n - \mathbf{U}_j^n)\otimes \mathbf{V}_j^n]\frac{\mathbf{c}_{ij}^n}{2d_{ij}} + \frac{1}{2}(\mathbf{U}_j^n + \mathbf{U}_i^n)\right\} \\ &= [1 + \frac{2\Delta t^n d_{ii}}{m_i^{n+1}}]\mathbf{U}_i^n \\ &\quad - \sum_{\substack{j\neq i \\ j\in\mathcal{I}(S_i^n)}} \frac{2\Delta t^n d_{ij}}{m_i^{n+1}} \left\{[\mathbf{F}_j^n - \mathbf{F}_i^n + (\mathbf{U}_i^n - \mathbf{U}_j^n)\otimes \mathbf{V}_j^n]\mathbf{n}_{ij}^n\frac{\|\mathbf{c}_{ij}^n\|}{2d_{ij}} + \frac{1}{2}(\mathbf{U}_j^n + \mathbf{U}_i^n)\right\} \\ &= (1 + \frac{2\Delta t^n d_{ii}}{m_i^{n+1}})\mathbf{U}_i^n + \sum_{\substack{j\neq i \\ j\in\mathcal{I}(S_i^n)}} \frac{2\Delta t^n d_{ij}}{m_i^{n+1}}\widehat{\mathbf{U}}(\mathbf{U}_i^n, \mathbf{U}_j^n), \end{aligned} \quad (4.75)$$

where $\mathbf{n}_{ij}^n \in \mathcal{S}^{d-1}$ is the unit direction vector of $\mathbf{c}_{ij}^n$ and $\widehat{\mathbf{U}}(\mathbf{U}_i^n, \mathbf{U}_j^n)$ is defined by

$$\widehat{\mathbf{U}}(\mathbf{U}_i^n, \mathbf{U}_j^n) := [\mathbf{F}_i^n - \mathbf{F}_j^n + (\mathbf{U}_j^n - \mathbf{U}_i^n)\otimes \mathbf{V}_j^n]\mathbf{n}_{ij}^n\frac{\|\mathbf{c}_{ij}^n\|}{2d_{ij}} + \frac{1}{2}(\mathbf{U}_j^n + \mathbf{U}_i^n). \quad (4.76)$$

Using the notation $\mathbf{v}$ and $\lambda_{\max}$ introduced in (4.64) one can see that

$$\widehat{\mathbf{U}}(\mathbf{U}_i^n, \mathbf{U}_j^n) = \int_{-1/2}^{1/2} \mathbf{v}(\mathbf{g}_j^n, \mathbf{n}_{ij}^n, \mathbf{U}_i^n, \mathbf{U}_j^n)(\bar{t}, x)\,\mathrm{d}x \quad (4.77)$$

where $\mathbf{g}_j^n$ is defined in (4.68) and

$$\bar{t} := \frac{\|\mathbf{c}_{ij}^n\|}{2d_{ij}}. \quad (4.78)$$

ince the special choice of d_{ij} as in (4.67) implies

$$\bar{t}\lambda_{\max}(\mathbf{g}_j^n, \mathbf{n}_{ij}^n, \mathbf{U}_i^n, \mathbf{U}_j^n) \leq \frac{1}{2}, \tag{4.79}$$

and A is an invariant set (see Remark 4.1.2), it implies that $\widehat{\mathbf{U}}(\mathbf{U}_i^n, \mathbf{U}_j^n) \in A$.

From (4.75), since $d_{ij} \geq 0$ for $j \neq i$ and $\sum_j d_{ij} = 0$, one can see that in order to show $\mathbf{U}_i^{n+1} \in A$, one sufficient condition is to prove

$$\mathcal{Y} := 1 + \frac{2\Delta t^n d_{ii}}{m_i^{n+1}} \geq 0.$$

Since $\widehat{\theta}_i \geq 0$ and by the definition of $h_i^{n,l}$ in (4.69), we obtain that

$$\|\mathbf{c}_{ij}^{n,l}\|_{\ell^2} \leq \int_{S_{ij}(t_l^n)} \|\nabla\psi_j(\mathbf{x}, t_l^n)\|_{\ell^2}\psi_i^{n,l}(\mathbf{x})\,\mathrm{d}\mathbf{x} \leq [h_i^{n,l}]^{-1}\int_{S_{ij}(t_l^n)} \psi_i^{n,l}(\mathbf{x})\,\mathrm{d}\mathbf{x},$$

which implies that

$$\begin{aligned}
\sum_{\substack{j\neq i \\ j\in\mathcal{I}(S_i^n)}} d_{ij} \leq & \lambda_i^n \sum_{\substack{j\neq i \\ j\in\mathcal{I}(S_i^n)}} \|\mathbf{c}_{ij}^n\|_{\ell^2} \\
\leq & \lambda_i^n \sum_{\substack{j\neq i \\ j\in\mathcal{I}(S_i^n)}} \sum_{l\in\mathcal{L}} \omega_l \|\mathbf{c}_{ij}^{n,l}\|_{\ell^2} \\
\leq & \lambda_i^n \sum_{\substack{j\neq i \\ j\in\mathcal{I}(S_i^n)}} \sum_{l\in\mathcal{L}} [h_i^{n,l}]^{-1}\omega_l \int_{S_{ij}(t_l^n)} \psi_i^{n,l}(\mathbf{x})\,\mathrm{d}\mathbf{x} \\
= & \lambda_i^n [h_i^n]^{-1}\kappa_i^n \sum_{l\in\mathcal{L}} \omega_l \int_{\Omega^{n,l}} \psi_i^{n,l}(\mathbf{x})\,\mathrm{d}\mathbf{x} \\
\leq & \lambda_i^n [h^n]^{-1}\kappa^n \sum_{l\in\mathcal{L}} \omega_l \int_{\Omega^{n,l}} \psi_i^{n,l}(\mathbf{x})\,\mathrm{d}\mathbf{x}
\end{aligned}$$

Since $d_{ii} = -\sum_{j\in\mathcal{I}(S_i^n)\backslash\{i\}} d_{ij}$, the condition implies that

$$\begin{aligned}\mathcal{Y} =& 1 - 2\Delta t^n \sum_{\substack{j\neq i \\ j\in\mathcal{I}(S_i^n)}} \frac{d_{ij}}{m_i^{n+1}} \\ \geq& 1 - 2\Delta t^n (h^n)^{-1}\kappa^n\lambda_i^n \frac{\sum_{l\in\mathcal{L}}\omega_l \int_{\Omega^{n,l}} \psi_i^{n,l}(\mathbf{x})\,\mathrm{d}\mathbf{x}}{m_i^{n+1}} \geq 0,\end{aligned}$$

which completes the proof. □

Remark 4.6.10. *Since* $\kappa_i^n \leq \mathrm{card}\mathcal{I}(S_i) - 1$, *we see that*

$$\kappa^n \leq \max_j \mathrm{card}\mathcal{I}(S_j) - 1. \tag{4.80}$$

Remark 4.6.11. *The assumption* $m_i^{n+1} > 0$ *is satisfied by choosing small* Δt^n *since* $m_i^n > 0$. *It implies that the mesh velocity* $\mathbf{V}^n$ *can not be chosen arbitrarily since* (4.63).

Corollary 4.6.12. *Let* $A \subset \mathcal{A}_{\mathbf{F}}$ *be an invariant set of* (4.1) *as defined in Definition 4.1.1 and* d_{ij} *chosen as in* (4.67). *For any cell* K, *under the same CFL condition* (4.72), *if*

$$\{\mathbf{U}_j^n : \exists i \in \mathcal{I}(K)\, s.t.\, j \in \mathcal{I}(S_i)\} \subset A,$$

then

$$\mathbf{u}_h^{n+1}(\mathbf{x}) \in A, \quad \forall \mathbf{x} \in K. \tag{4.81}$$

In particular, if $\{\mathbf{U}_i^n : \forall i\} \subset A$, *then* $\mathbf{u}_h^{n+1}(\mathbf{x}) \in A$ *for any* $\mathbf{x}$.

Proof. Since

$$\{\mathbf{U}_j^n : \exists i \in \mathcal{I}(K) s.t. j \in \mathcal{I}(S_i)\} \subset A,$$

it follows that

$$\{\mathbf{U}_i^{n+1} : \exists i \in \mathcal{I}(K)\} \subset A.$$

Since A is a convex set, using Lemma 4.4.1, we conclude that $\mathbf{u}_h^{n+1}|_K \in A$. □

4.6.5 Discrete entropy inequality

Lemma 4.6.13. *Let $(\eta, \mathbf{q})$ be an entropy pair of (4.1). Then $(\eta(\mathbf{v}), \mathbf{q}(\mathbf{v}) - \eta(\mathbf{v})\mathbf{W})$ is an entropy pair of the problem $\partial_t \mathbf{v} + \nabla \cdot \mathbf{g}(\mathbf{v}) = 0$ with $\mathbf{g}(\mathbf{v}) = \mathbf{F}(\mathbf{v}) - \mathbf{v} \otimes \mathbf{W}$.*

Proof. By definition of entropy pair in (4.4), we need to show that for $\forall \mathbf{n} \in \mathcal{S}^{d-1}$

$$\partial_{v_k}([\mathbf{q}(\mathbf{v}) - \eta(\mathbf{v})\mathbf{W}]\cdot\mathbf{n}) = \sum_{i=1}^{m}\sum_{j=1}^{d} \partial_{v_i}\eta(\mathbf{v})\partial_{v_k}(\mathbf{g}_{ij}(\mathbf{v})n_j).$$

This is true Since

$$\begin{aligned}
\sum_{i=1}^{m}\sum_{j=1}^{d} \partial_{v_i}\eta(\mathbf{v})\partial_{v_k}(\mathbf{g}_{ij}(\mathbf{v})n_j) =& \sum_{i=1}^{m}\sum_{j=1}^{d} \partial_{v_i}\eta(\mathbf{v})\partial_{v_k}(\mathbf{F}_{ij}(\mathbf{v})n_j) - \sum_{i=1}^{m}\sum_{j=1}^{d} \partial_{v_i}\eta(\mathbf{v})\partial_{v_k}(v_i W_j n_j) \\
=& \partial_{v_k}(\mathbf{q}(\mathbf{v})\cdot\mathbf{n}) - \sum_{i=1}^{m} \partial_{v_i}\eta(\mathbf{v})\delta_{ik}\mathbf{W}\cdot\mathbf{n}) \\
=& \partial_{v_k}(\mathbf{q}(\mathbf{v})\cdot\mathbf{n}) - \partial_{v_k}\eta(\mathbf{v})\mathbf{W}\cdot\mathbf{n} \\
=& \partial_{v_k}[(\mathbf{q}(\mathbf{v}) - \eta(\mathbf{v})\mathbf{W}]\cdot\mathbf{n},
\end{aligned}$$

where the assumption that $(\eta, \mathbf{q})$ is an entropy pair of (4.1) is used in the 2nd equality. □

Theorem 4.6.14 (Discrete Entropy Inequality)**.** *Let $(\eta, \mathbf{q})$ be a entropy pair of (4.1). Under the same CFL condition (4.72), the following discrete entropy inequality holds for the solution of the scheme (4.50)*

$$\begin{aligned}
&\frac{m_i^{n+1}\eta(\mathbf{U}^{n+1})_i - m_i^n\eta(\mathbf{U}_i^n)}{\Delta t^n} \\
&\quad + \sum_l \omega_l \int_{\Omega_{t^n,l}} \sum_j \nabla\cdot[(\mathbf{q}(\mathbf{U}_j^n) - \eta(\mathbf{U}_j^n)\mathbf{V}_j^n)\psi_j^{n,l}(\mathbf{x})]\psi_i^{n,l}(\mathbf{x})\,\mathrm{d}\mathbf{x} \\
&\quad + B(\sum_j \eta(\mathbf{U}_j^n)\psi_j^n, \psi_i^n) \le 0. \quad (4.82)
\end{aligned}$$

Proof. Since $\eta(\mathbf{u})$ is a convex function with respect to $\mathbf{u}$ and the CFL condition (4.72) is

satisfied, from (4.75), it follows that

$$\eta(\mathbf{U}_i^{n+1}) \leq \left(1 + \frac{2\Delta t^n d_{ii}}{m_i^{n+1}}\right)\eta(\mathbf{U}_i^n) + \sum_{\substack{j\neq i \\ j\in\mathcal{I}(S_i^n)}} \frac{2\Delta t^n d_{ij}}{m_i^{n+1}}\eta(\widehat{\mathbf{U}}(\mathbf{U}_i^n, \mathbf{U}_j^n)). \tag{4.83}$$

From (4.77), we see that $\widehat{\mathbf{U}}(\mathbf{U}_i^n, \mathbf{U}_j^n)$ is the average of the solution $\mathbf{v}(\mathbf{g}_j^n, \mathbf{n}_{ij}^n, \mathbf{U}_i^n, \mathbf{U}_j^n)(\bar{t}, x)$ with

$$\bar{t} := \frac{\|\mathbf{c}_{ij}^n\|}{2d_{ij}}.$$

By Lemma 4.6.13, it follows that $(\eta(\mathbf{v}), \mathbf{q}(\mathbf{v}) - \eta(\mathbf{v})\mathbf{V}_j^n)$ is a entropy pair of the problem $\partial_t\mathbf{v} + \nabla\cdot\mathbf{g}(\mathbf{v}) = 0$ with $\mathbf{g}(\mathbf{v}) = \mathbf{F}(\mathbf{v}) - \mathbf{v}\otimes\mathbf{V}_j^n$. From (4.76), we obtain that

$$\begin{aligned}\eta(\widehat{\mathbf{U}}(\mathbf{U}_i^n, \mathbf{U}_j^n)) \leq \frac{1}{2}[\eta(\mathbf{U}_i^n) + \eta(\mathbf{U}_j)] \\ - \bar{t}\left\{[\mathbf{q}(\mathbf{U}_j) - \eta(\mathbf{U}_j)\mathbf{V}_j^n]\cdot\mathbf{n}_{ij}^n - [\mathbf{q}(\mathbf{U}_i^n) - \eta(\mathbf{U}_i^n)\mathbf{V}_j^n]\cdot\mathbf{n}_{ij}^n]\right\}.\end{aligned} \tag{4.84}$$

Plugging (4.84) into (4.83), we have

$$\begin{aligned}\frac{m_i^{n+1}}{\Delta t^n}[\eta(\mathbf{U}_i^{n+1}) - \eta(\mathbf{U}_i^n)] &\leq 2d_{ii}\eta(\mathbf{U}_i^{n+1}) + \sum_{\substack{j\neq i \\ j\in\mathcal{I}(S_i^n)}} d_{ij}[\eta(\mathbf{U}_i^n) + \eta(\mathbf{U}_j)] \\ &\quad - \sum_{\substack{j\neq i \\ j\in\mathcal{I}(S_i^n)}} \left\{[\mathbf{q}(\mathbf{U}_j) - \eta(\mathbf{U}_j)\mathbf{V}_j^n]\cdot\mathbf{c}_{ij}^n - [\mathbf{q}(\mathbf{U}_i^n) - \eta(\mathbf{U}_i^n)\mathbf{V}_j^n]\cdot\mathbf{c}_{ij}^n]\right\} \\ &= \sum_{j\in\mathcal{I}(S_i^n)} d_{ij}\eta(\mathbf{U}_j^n) - \sum_{j\in\mathcal{I}(S_i^n)} [\mathbf{q}(\mathbf{U}_j) - \eta(\mathbf{U}_j)\mathbf{V}_j^n]\cdot\mathbf{c}_{ij}^n - \sum_{\substack{j\neq i \\ j\in\mathcal{I}(S_i^n)}} \eta(\mathbf{U}_i^n)(\mathbf{V}_i^n - \mathbf{V}_j^n)\mathbf{c}_{ij}^n \\ &= \sum_{j\in\mathcal{I}(S_i^n)} d_{ij}\eta(\mathbf{U}_j^n) - \sum_{j\in\mathcal{I}(S_i^n)} [\mathbf{q}(\mathbf{U}_j) - \eta(\mathbf{U}_j)\mathbf{V}_j^n]\cdot\mathbf{c}_{ij}^n + \sum_{j\in\mathcal{I}(S_i^n)} \eta(\mathbf{U}_i^n)\mathbf{V}_j^n\mathbf{c}_{ij}^n\end{aligned}$$

From (4.63), we conclude that

$$\frac{m_i^{n+1}\eta(\mathbf{U}_i^{n+1}) - m_i^n\eta(\mathbf{U}_i^n)}{\Delta t^n} \leq \sum_{j\in\mathcal{I}(S_i^n)} d_{ij}\eta(\mathbf{U}_j^n) - \sum_{j\in\mathcal{I}(S_i^n)} [\mathbf{q}(\mathbf{U}_j) - \eta(\mathbf{U}_j)]\mathbf{V}_j^n\cdot\mathbf{c}_{ij}^n,$$

which implies (4.82) by using the definition of d_{ij} in (4.67) and the definition of $\mathbf{c}_{ij}^n$ in (4.55).

□

4.6.6 Piecewise viscosity

If the piecewise viscosity ν_K is used as in (3.8), the bilinear form B in (4.50) becomes

$$B(\mathbf{u}_h^n, \psi_i^n) = \sum_{j \in \mathcal{I}(S_i^n)} B(\psi_j^n, \psi_i^n) \mathbf{U}_j^n \tag{4.85}$$

and

$$B(\psi_j^n, \psi_i^n) = \sum_{K \in S_{ij}^n} \nu_K^n b_K(\psi_j^n, \psi_i^n),$$

where the bilinear forms b_K are assumed to satisfy all the properties of Definition 3.3.1.

Define

$$d_{ij} = \sum_{K \in S_{ij}^n} \nu_K^n b_K(\psi_j^n, \psi_i^n), \quad j \neq i. \tag{4.86}$$

The scheme (4.50) with numerical viscosity (4.85) is equal to

$$m_i^{n+1} \mathbf{U}_i^{n+1} - m_i^n \mathbf{U}_i^{n+1} + \Delta t^n \sum_{j \in \mathcal{I}(S_i^n)} [\mathbf{F}_j^n - \mathbf{U}_j^n \otimes \mathbf{V}_j^n] \mathbf{c}_{ij}^n - \sum_{j \in \mathcal{I}(S_i^n)} d_{ij} \mathbf{U}_j^n = 0, \quad \forall i, \tag{4.87}$$

which is exactly equal to the scheme (4.60) with d_{ij} defined in (4.86).

Theorem 4.6.1, Lemma 4.6.3, and Theorem 4.6.6 still hold for this new algorithm (4.87). Theorem 4.6.9 also holds by choosing

$$\nu_K^n := \max_{\substack{i,j \in \mathcal{I}(K) \\ i \neq j}} \frac{\|\mathbf{c}_{ij}^n\|_{\ell^2} \lambda_{\max}(\mathbf{g}_j^n, \mathbf{n}_{ij}^n, \mathbf{U}_i^n, \mathbf{U}_j^n)}{-b_K(\psi_j^n, \psi_i^n)} \tag{4.88}$$

under certain CFL condition. This is because of

$$d_{ij} = \sum_{K \subset S_{ij}^n} \nu_K b_K(\psi_j^n, \psi_i^n) \geq \sum_{K \subset S_{ij}^n} \|\mathbf{c}_{ij}^n\|_{\ell^2} \lambda_{\max}(\mathbf{g}_j^n, \mathbf{n}_{ij}^n, \mathbf{U}_i^n, \mathbf{U}_j^n)$$

which implies that (4.79). For $d = 2$, since each edge is shared by two cells (without considering boundary condition) ν_K^n can be chosen as

$$\nu_K^n := \max_{\substack{i,j \in \mathcal{I}(K) \\ i \neq j}} \frac{\|\mathbf{c}_{ij}^n\|_{\ell^2} \lambda_{\max}(\mathbf{g}_j^n, \mathbf{n}_{ij}^n, \mathbf{U}_i^n, \mathbf{U}_j^n)}{-2b_K(\psi_j^n, \psi_i^n)}.$$

4.6.7 New schemes with its extension to SSPRK methods

Recall that (3.13) or (3.14) are two examples of SSPRk methods which can be used to solve the ODE system $\frac{d\mathbf{u}}{dt} = L(\mathbf{u})$ by using the forward Euler method in sub-steps. For the ALE method (4.50) or (4.60), the main variables are $\{m_i^n, \mathbf{U}_i^n, \mathbf{a}_i^n\}$. We have to choose the independent variables first in order to use a SSPRK method directly. Since it is easier to compute m_i^n from $\{\mathbf{a}_j^n\}$, it is better to choose $\{\mathbf{a}_j^n\}$ as part of the independent of variables. However, maybe the combination like $\{m_i^n \mathbf{u}_i^n\}$ should be chosen as independent variables to make the whole SSPRK method keep the conservation property and the invariant domain property. But, this is not obvious. Let us consider SSPRK2 (3.13). Assume the solutions obtained by using the forward Euler method in substeps are $\{m_i^{n+1,1}, \mathbf{U}_i^{n+1,1}, \mathbf{a}_i^{n+1,1}\}$ and $\{m_i^{n+1,2}, \mathbf{U}_i^{n+1,2}, \mathbf{a}_i^{n+1,2}\}$. To get the solution at t^{n+1}, if $\{m_i^n \mathbf{U}_i^n, \mathbf{a}_i^n\}$ are chosen as independent variables, then we first get $\mathbf{a}_i^{n+1} = \frac{1}{2}\mathbf{a}_i^{n+1,1} + \frac{1}{2}\mathbf{a}_i^{n+1,2}$ and then compute m_i^{n+1}. From $m_i^{n+1}\mathbf{U}_i^{n+1} = \frac{1}{2}m_i^{n+1,1}\mathbf{U}_i^{n+1,1} + \frac{1}{2}m_i^{n+1,2}\mathbf{U}_i^{n+1,2}$, we can compute $\mathbf{U}_i^{n+1}$. The method is conservative since $\int \mathbf{u}_h^{n+1} = \sum_i m_i^{n+1}\mathbf{U}_i^{n+1}$. However, $\mathbf{U}_i^{n+1}$ is not necessary the convex combination of $\mathbf{U}_i^{n+1,1}$ and $\mathbf{U}_i^{n+1,2}$ and the method may not keep the invariant domain property. If $\{\mathbf{U}_i^n, \mathbf{a}_i^n\}$ are chosen as independent variables, then the invariant domain property is preserved, but the conservation may be not. It looks like those two properties, conservation and invariant domain property, are not compatible with each other.

One method to get SSPRK extension is to relax the conservation property and to keep

discrete conservation only. The method is as follows

$$\frac{\mathfrak{m}_i^{n+1}\mathbf{U}_i^{n+1}-\mathfrak{m}_i^n\mathbf{U}_i^n}{\Delta t^n}=\sum_{j\in\mathcal{I}(S_i^n)}d_{ij}^n\mathbf{U}_j^n \\ -\int_{\Omega_{t^n}}\nabla\cdot\Big(\sum_j(\mathbf{F}(\mathbf{U}_j^n)-\mathbf{U}_j^n\otimes\mathbf{V}_j^n)\psi_j^n(\mathbf{x})\Big)\psi_i^n(\mathbf{x})\,\mathrm{d}\mathbf{x}, \quad (4.89)$$

where

$$\mathfrak{m}_i^0=m_i^0,$$

$$\mathfrak{m}_i^{n+1}=\mathfrak{m}_i^n+\Delta t^n\int_{S_i^n}\psi_i^n(\mathbf{x})\nabla\cdot\mathbf{V}_h^n(\mathbf{x})\,\mathrm{d}\mathbf{x}, \quad (4.90)$$

and $\mathbf{V}_h^n=\sum_j\mathbf{V}_j^n\psi_j\in\mathbf{P}_d(\mathcal{T}_h^n)$.

Remark 4.6.15. *Comparing* (4.63) *and* (4.90)*, we see that* $\mathfrak{m}_i^{n+1}\neq m_i^{n+1}$ *in general.*

Remark 4.6.16. *In the scheme* (4.89) *or* (4.92)*, the assumption on* $\mathbf{V}_h^n$ *is* $\mathbf{V}_h^n\in\mathbf{P}_d(\mathcal{T}_h^n)$*. We do not need to assume* $\mathbf{V}_h^n\in\widehat{P}^{\mathrm{geo}}\subset\widehat{P}$*. Moreover, we do not need to use any quadrature rule because of Lemma 4.6.19.*

Define $\mathbf{c}_{ij}^n$ as

$$\mathbf{c}_{ij}^n:=\int_{S_i^n}\nabla\psi_j^n(\mathbf{x})\psi_i^n(\mathbf{x})\,\mathrm{d}\mathbf{x}. \quad (4.91)$$

We obtain an equivalent form of the scheme (4.89) as follows

$$\frac{\mathfrak{m}_i^{n+1}\mathbf{U}_i^{n+1}-\mathfrak{m}_i^n\mathbf{U}_i^n}{\Delta t}+\sum_{j\in\mathcal{I}(S_i^n)}(\mathbf{F}(\mathbf{U}_j^n)-\mathbf{U}_j^n\otimes\mathbf{V}_j^n)\cdot\mathbf{c}_{ij}^n-d_{ij}^n\mathbf{U}_j^n=0. \quad (4.92)$$

In addition, we have

$$\mathfrak{m}_i^{n+1}=\mathfrak{m}_i^n+\Delta t^n\sum_{j\in\mathcal{I}(S_i^n)}\mathbf{V}_j^n\cdot\mathbf{c}_{ij}^n. \quad (4.93)$$

Compared to (4.50), the term $\mathbf{c}_{ij}^n$ in (4.92) is defined in (4.91) and we do not need to use a quadrature rule for time integral.

Definition 4.6.17. *The method is discrete conservative if* $\sum_i \mathfrak{m}_i^{n+1}\mathbf{U}_i^{n+1} = \sum_i \mathfrak{m}_i^n \mathbf{U}_i^n$ *for any* n.

Theorem 4.6.18. *The scheme* (4.89) *or* (4.92) *is discrete conservative.*

Proof. The conclusion follows from the fact that $\sum_i \mathbf{c}_{ij}^n = 0$ and $\sum_i d_{ij}^n = 0$. □

The scheme (4.89) or (4.92) satisfies DGCL.

Theorem 4.6.19 (DGCL)**.** *If* $\mathbf{U}_j^n = \mathbf{C} \in \mathbb{R}^m$ *for all* $j \in \mathcal{I}(S_i^n)$*, then the solution of the scheme* (4.89) *or* (4.92) *satisfies that* $\mathbf{U}_i^{n+1} = \mathbf{C}$*. In particular, it implies that the scheme* (4.89) *preserve constant states.*

Proof. The conclusion follows from the relation (4.93) and (4.92). □

Define

$$h^n := \min_i h_i^n, \quad h_i^n := \min_{j\in\mathcal{I}(S_i^n)} \frac{1}{\|\|\nabla\psi_j\|_{\ell^2}\|_{L^\infty(S_{ij}^n)}}. \tag{4.94}$$

Define

$$\kappa^n := \max_i \kappa_i^n, \quad \kappa_i^n := \frac{\sum_{j\in\mathcal{I}(S_i^n)\setminus\{i\}} \int_{S_{ij}^n} \psi_i^n(\mathbf{x})\,\mathrm{d}\mathbf{x}}{\int_{\Omega^n} \psi_i^n(\mathbf{x})\,\mathrm{d}\mathbf{x}}. \tag{4.95}$$

Theorem 4.6.20 (Invariant Domain Property)**.** *Let* $A \subset \mathcal{A}_{\mathbf{F}}$ *be an invariant set of* (4.1) *as defined in Definition 4.1.1 and* d_{ij} *is chosen as in* (4.67) *with* $\mathbf{c}_{ij}^n$ *defined in* (4.91)*. Assume* $\mathfrak{m}_i^{n+1} > 0$ *for all* i*. If* $\{\mathbf{U}_j^n : j \in \mathcal{I}(S_i^n)\} \subset A$ *and* Δt^n *satisfies the following CFL condition*

$$2\Delta t^n \frac{\kappa^n \max_i \lambda_i^n}{h^n} \max_i \frac{m_i^n}{\mathfrak{m}_i^{n+1}} \le 1, \tag{4.96}$$

where h^n *and* κ^n *are defined in* (4.94) *and* (4.95)*, then the solution of the scheme* (4.89) *or* (4.92) *has the property that* $\mathbf{U}_i^{n+1} \in A$*. In particular, if* $\{\mathbf{U}_j^n : \forall j\} \subset A$*, then* $\{\mathbf{U}_j^{n+1} : \forall j\} \subset A$.

Proof. The proof is the same as the proof of Theorem 4.6.9 since the relations (4.75) and (4.65) hold also with $\mathbf{c}_{ij}^n$ defined in (4.91). □

Theorem 4.6.21 (Discrete Entropy Inequality)**.** *Let* $(\eta, \mathbf{q})$ *be a entropy pair of* (4.1)*. Under the same CFL condition* (4.96)*, the following discrete entropy inequality holds for the solution of the scheme* (4.89)

$$\frac{\mathfrak{m}_i^{n+1}\mathbf{U}_i^{n+1} - \mathfrak{m}_i^n\mathbf{U}_i^n}{\Delta t^n} + \int_{\Omega_{t^n}} \sum_j \nabla\cdot[(\mathbf{q}(\mathbf{U}_j^n) - \eta(\mathbf{U}_j^n)\mathbf{V}_j^n)\psi_j^n(\mathbf{x})]\psi_i^n(\mathbf{x})\,\mathrm{d}\mathbf{x} + B(\sum_j \eta(\mathbf{U}_j^n)\psi_j^n, \psi_i^n) \le 0. \quad (4.97)$$

The scheme (4.89) is important since it can be extended to higher order SSPRK methods. The three stage third order SSPRK3 (3.14) is implemented as in Algorithm 4.

Algorithm 4 SPP RK3: Get $\mathbf{u}_h^1$ from $\mathbf{u}_h^0$

Require: $\mathcal{T}_h^0$, $\mathbf{u}_h^0$, $\mathfrak{m}^0$, t^0

1: Choose $\mathbf{V}_h^0$; Call Euler step($\mathcal{T}_h^0$, $\mathbf{u}_h^0$, $\mathfrak{m}^0$, $\mathbf{V}_h^0$, Δt^1, $\mathcal{T}_h^1$, $\mathbf{u}_h^1$, $\mathfrak{m}^1$)

2: Choose $\mathbf{V}_h^1$; Call Euler step($\mathcal{T}_h^1$, $\mathbf{u}_h^1$, $\mathfrak{m}^1$, $\mathbf{V}_h^1$, Δt^1, $\tilde{\mathcal{T}}_h^2$, $\tilde{\mathbf{u}}_h^2$, $\tilde{\mathfrak{m}}^2$)

3: Get $\mathcal{T}_h^2$ by $\mathbf{a}^2 = \frac{3}{4}\mathbf{a}^0 + \frac{1}{4}\tilde{\mathbf{a}}^2$. Set $\mathfrak{m}^2 = \frac{3}{4}\mathfrak{m}^0 + \frac{1}{4}\tilde{\mathfrak{m}}^2$. Get $\mathbf{u}_h^2 = \frac{3}{4}\frac{\mathfrak{m}^0}{\mathfrak{m}^2}\mathbf{u}_h^0 + \frac{1}{4}\frac{\tilde{\mathfrak{m}}^2}{\mathfrak{m}^2}\tilde{\mathbf{u}}_h^2$.

4: Choose $\mathbf{V}_h^2$. Call Euler step($\mathcal{T}_h^2$, $\mathbf{u}_h^2$, $\mathfrak{m}^2$, $\mathbf{V}_h^2$, Δt^1, $\tilde{\mathcal{T}}_h^3$, $\tilde{\mathbf{u}}_h^3$, $\tilde{\mathfrak{m}}^3$)

5: Get $\mathcal{T}_h^3$ by $\mathbf{a}^3 = \frac{1}{3}\mathbf{a}^0 + \frac{2}{3}\tilde{\mathbf{a}}^3$. Set $\mathfrak{m}^3 = \frac{1}{3}\mathfrak{m}^0 + \frac{2}{3}\tilde{\mathfrak{m}}^3$. Get $\mathbf{u}_h^2 = \frac{1}{3}\frac{\mathfrak{m}^0}{\mathfrak{m}^3}\mathbf{u}_h^0 + \frac{2}{3}\frac{\tilde{\mathfrak{m}}^3}{\mathfrak{m}^3}\tilde{\mathbf{u}}_h^3$.

6: **return** $\mathcal{T}_h^3$, $\mathbf{u}_h^3$, $\mathfrak{m}^3$, $t^1 = t^0 + \Delta t^1$.

Theorem 4.6.22. *The Algorithm 4 is discrete conservative and satisfies the invariant domain property under some CFL condition.*

Proof. Suppose A is the invariant domain of the problem. And $\mathbf{U}_i^0 \in A$ for all i. Let $\mathbf{u}_h^1 = \sum_j \mathbf{U}_i^1\psi_i^1$ and $\tilde{\mathbf{u}}_h^2 = \sum_j \tilde{\mathbf{U}}_i^1\psi_i^2$ where $\mathbf{u}_h^1$ and $\tilde{\mathbf{u}}_h^2$ are obtained in Step-1 and Step-2 of the Algorithm 4 and ψ_i^1 and ψ_i^2 are the basis related to the meshes $\mathcal{T}_h^1$ and $\tilde{\mathcal{T}}_h^2$. By

Theorem 4.6.18, we obtain that $\sum_i \mathfrak{m}_i^1 \mathbf{U}_i^1 = \sum_i \mathfrak{m}_i^0 \mathbf{U}_i^0$ and $\sum_i \widetilde{\mathfrak{m}}_i^2 \tilde{\mathbf{U}}_i^1 = \sum_i \mathfrak{m}_i^1 \mathbf{U}_i^1$. It follows that $\sum_i \widetilde{\mathfrak{m}}_i^2 \tilde{\mathbf{U}}_i^1 = \sum_i \mathfrak{m}_i^0 \mathbf{U}_i^0$. In Step-3 of the Algorithm 4, we see that

$$\sum_i \mathfrak{m}_i^2 \mathbf{U}_i^2 = \frac{3}{4} \sum_i \mathfrak{m}_i^0 \mathbf{U}_i^0 + \frac{1}{4} \sum_i \widetilde{\mathfrak{m}}_i^2 \tilde{\mathbf{U}}_i^2 = \sum_i \mathfrak{m}_i^0 \mathbf{U}_i^0.$$

Since the forward Euler method has the invariant domain property, it follows that $\mathbf{U}_i^1 \in A$ and $\tilde{\mathbf{U}}_i^1 \in A$ for all i. Since $\mathbf{u}_h^2 = \frac{3}{4}\frac{\mathfrak{m}^0}{\mathfrak{m}^2}\mathbf{u}_h^0 + \frac{1}{4}\frac{\widetilde{\mathfrak{m}}^2}{\mathfrak{m}^2}\tilde{\mathbf{u}}_h^2$ and $\mathfrak{m}^2 = \frac{3}{4}\mathfrak{m}^0 + \frac{1}{4}\tilde{\mathfrak{m}}^2$, we see that $\mathbf{U}_i^2$ is a convex combination of $\mathbf{U}_i^0$ and $\tilde{\mathbf{U}}_i^2$. Since A is convex, we conclude that $\mathbf{U}_i^2 \in A$ for all i. Likewise, one can get that $\sum_i \mathfrak{m}_i^3 \mathbf{U}_i^3 = \sum_i \mathfrak{m}_i^0 \mathbf{U}_i^0$ and $\mathbf{U}_i^3 \in A$ for all i, which complete the proof. □

4.6.8 Algorithm on choosing V_h^n

Many techniques have been proposed in the literature. In [26], the method is to model the deformation of the whole initial domain by considering it as a "elastic" solid, see (4.5) and (4.6) of [26]. In [79], several mesh moving strategies are mentioned, including tension spring analogy, torsion spring analogy, truss analogy and linear elasticity analogy. In (7) of [77], an elliptic problem is used to get the mesh velocity or ALE velocity, where the monitor function $M(x,t)$ is user-defined and is chosen as the density ρ or $1+\alpha|\nabla\rho|^2$ for the Euler equations in their numerical tests.

Because the ALE method is designed to combine the advantages of the Eulerian method and the Lagrangian method, it is desirable that the mesh velocity $\mathbf{V}_h^n$ is close to the Lagrangian velocity or characteristic velocity for scalar conservation laws, and avoid severe mesh distortion. To find $\mathbf{V}$ is equivalent to find the location of vertices $\mathbf{a}^{n+1}$ at the next time step t^{n+1} since $\mathbf{V}_i^n = \frac{\mathbf{a}_i^{n+1} - \mathbf{a}_i^n}{\Delta t^{n+1}}$. Let $\mathcal{T}_{\mathrm{Lg}}^{n+1}$ denote the mesh which is obtained by using the Lagrangian method or the Characteristic method. Since in the mesh $\mathcal{T}_{\mathrm{Lg}}^{n+1}$ the area of some cell may be much small or the interior angle may be too big or too small, a good approximation is to smooth it and get $\mathcal{T}_{\mathrm{Sm}}^{n+1}$. The method we use to get $\mathcal{T}_{\mathrm{Sm}}^{n+1}$ is to move

each node $\mathbf{a}_i^{n+1}$ to the center of $\mathcal{S}_i$ by iteration L times. The algorithm is as follows

$$\begin{cases} \mathbf{a}_j^{n+1,0} := \mathbf{a}_{j,\mathrm{Lg}}^{n+1} \\ \mathbf{a}_i^{n+1,l} := \dfrac{1}{|\{j : i \neq j \in \mathcal{I}(\mathcal{S}_i)\}|} \displaystyle\sum_{i \neq j \in \mathcal{I}(\mathcal{S}_i)} \mathbf{a}_j^{n+1,l-1}, \quad l = 1 \dots L, \\ \mathbf{a}_{j,\mathrm{Sm}}^{n+1} := \mathbf{a}_j^{n+1,L}. \end{cases} \tag{4.98}$$

Then $\mathbf{a}^{n+1}$ is chosen as

$$\mathbf{a}^{n+1} = \omega \mathbf{a}_{\mathrm{Lg}}^{n+1} + (1-\omega)\mathbf{a}_{\mathrm{Sm}}^{n+1},$$

to get $\mathcal{T}_h^{n+1}$, where ω is a user-defined constant. In all computations, we use $\omega = 0.9$ and $L = 2$. The convex combination is used to avoid severe changes in some parts of mesh (which will decrease the time step extremely) so that the mesh is moving smoothly along the flow.

For a vertex of the boundary, for simplicity, its moving velocity is either the flow velocity, or the projection of the flow velocity to the tangent direction of the boundary, i.e., $\mathbf{v} \cdot \mathbf{n} = 0$.

One advantage of this method is that the mesh will only move in the x direction if the problem is a 1D problem no matter what dimension the mesh is. This property enable us to refine the mesh only in the x direction for 1D problems even on a 2D mesh. For example, we can apply this ALE method to solve the 1D Burgers equation on a 2D mesh. The above method is similar to the method used in [63] where a similar technique is used to move the "generator" in the rezone phase. As is mentioned in [63], a more advanced method is to choose ω point-wise by using the right Cauchy-Green strain tensor (see, e.g., [51, (2.37)]).

In [67], several techniques are introduced about which nodes should be moved, and how to move those nodes. Whether a vertex is labeled to be moved depends on the distortion of the local mesh which is determined by applying an angle test and a volume test. The angle test is used to quantify the shear distortion while the volume or area test is for volumetric

distortion (see [2]). For mesh smoothing, two methods are mentioned there. The first one is based on Winslow's work [78] with equipotential smoothing; the other one is Tipton's variational method which is based on reformulating Winslow's method. Other techniques on grid generation (see, e.g. [68, §10.3][52, §5.4.2]) can be used to get new mesh at each time step. The method used here is to move the vertex to the averaging locations of the surrounding nodes, which is also mentioned in [2].

4.6.9 Application to Euler equations

4.6.9.1 Computing $\lambda_{\max}$

The 2D Euler equations can be written as

$$\partial_t \mathbf{u} + \partial_x \mathfrak{F}(\mathbf{u}) + \partial_y \mathfrak{G}(\mathbf{u}) = 0, \tag{4.99}$$

where

$$\mathbf{u} = (\rho, \rho u, \rho v, E)^{\mathsf{T}},$$

$$\mathfrak{F} = (\rho u, \rho u^2 + p, \rho uv, u(E+p))^{\mathsf{T}},$$
$$\mathfrak{G} = (\rho v, \rho uv, \rho v^2 + p, v(E+p))^{\mathsf{T}},$$

p is the pressure, e is the special internal energy, and E is the total energy. For a polytropic ideal gas, the equation of state is

$$p = (\gamma - 1)\rho e$$

with $1 < \gamma \leq \frac{5}{3}$, (see, e.g., [31, §1.2]).

Lemma 4.6.23 (Rotation invariant property [74, Proposition 3.15])**.** *For fixed θ, the following rotational invariant property holds*

$$\cos\theta \mathfrak{F}(\mathbf{u}) + \sin\theta \mathfrak{G}(\mathbf{u}) = \mathbf{T}(\theta)^{-1} \mathfrak{F}(\mathbf{T}(\theta)\mathbf{u}), \tag{4.100}$$

where

$$\mathbf{T}(\theta)=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & \cos\theta & \sin\theta & 0\\ 0 & -\sin\theta & \cos\theta & 0\\ 0 & 0 & 0 & 1\end{bmatrix}\quad \mathbf{T}(\theta)^{-1}=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & \cos\theta & -\sin\theta & 0\\ 0 & \sin\theta & \cos\theta & 0\\ 0 & 0 & 0 & 1\end{bmatrix}. \tag{4.101}$$

Remark 4.6.24. *The above rotation invariant property holds also for the 3D case, see [74, Proposition 3.19].*

By definition of $T(\theta)$ in (4.101), it is easy to see that if $\mathbf{u}_1=\mathbf{T}(\theta)\mathbf{u}_2$, then $\rho_1=\rho_2$, $e_1=e_2$, $E_1=E_2$ and $(u_1,v_1)^{\mathsf{T}}=\mathcal{R}_\theta(u_2,v_2)^{\mathsf{T}}$, where $\mathcal{R}_\theta$ mean the rotation in plane around center by θ clockwise.

Lemma 4.6.25. *The solution* $\mathbf{v}$ *of the Riemann problem*

$$\begin{cases}\partial_t\mathbf{v}+\partial_x\mathbf{g}_{ij}^n(\mathbf{v})=0,\\ \mathbf{v}(0,x)=\mathbf{U}_i^n\mathbb{1}_{x<0}+\mathbf{U}_j^n\mathbb{1}_{x>0}.\end{cases} \tag{4.102}$$

satisfies that

$$\mathbf{v}(t,x)=\mathbf{T}(\theta)^{-1}\mathbf{w}(t,(\mathbf{V}_j^n\cdot\mathbf{n}_{ij}^n)t+x), \tag{4.103}$$

where $\mathbf{g}_{ij}^n(\mathbf{v}):=(\mathbf{F}(\mathbf{v})-\mathbf{v}\otimes\mathbf{V}_j^n)\mathbf{n}_{ij}^n$, θ *is defined by* $\mathbf{n}_{ij}^n=(\cos\theta,\sin\theta)^{\mathsf{T}}$, *and* $\mathbf{w}$ *is the solution of the following 1D Riemann problem*

$$\begin{cases}\partial_t\mathbf{w}+\partial_x\mathfrak{F}(\mathbf{w})=0,\\ \mathbf{w}(0,x)=\mathbf{T}(\theta)\mathbf{U}_i^n\mathbb{1}_{x<0}+\mathbf{T}(\theta)\mathbf{U}_j^n\mathbb{1}_{x>0}.\end{cases} \tag{4.104}$$

Proof. By Lemma 4.6.23, the 1D Riemann problem (4.102) is equal to

$$\begin{cases} \partial_t \mathbf{v} + \partial_x[\mathbf{T}(\theta)^{-1}\mathfrak{F}(\mathbf{T}(\theta)\mathbf{v}) - (\mathbf{v}\otimes\mathbf{V}_j^n)\mathbf{n}_{ij}^n] = 0, \quad x\in[-\tfrac{1}{2},\tfrac{1}{2}], \\ \mathbf{v}(0,x) = \mathbf{U}_i^n \mathbb{1}_{x<0} + \mathbf{U}_j^n \mathbb{1}_{x>0}, \end{cases}$$

i.e.,

$$\begin{cases} \partial_t \mathbf{T}(\theta)\mathbf{v} + \partial_x[\mathfrak{F}(\mathbf{T}(\theta)\mathbf{v}) - (\mathbf{V}_j^n\cdot\mathbf{n}_{ij}^n)\mathbf{T}(\theta)\mathbf{v}] = 0, \\ \mathbf{v}(0,x) = \mathbf{U}_i^n \mathbb{1}_{x<0} + \mathbf{U}_j^n \mathbb{1}_{x>0}. \end{cases}$$

By changing variable as $\widehat{w} = \mathbf{T}(\theta)\mathbf{v}$, we see that $\widehat{w}$ satisfies the following equation

$$\begin{cases} \partial_t \widehat{w} + \partial_x[\mathfrak{F}(\widehat{w}) - (\mathbf{V}_j^n\cdot\mathbf{n}_{ij}^n)\widehat{w}] = 0, \\ \widehat{w}(0,x) = \mathbf{T}(\theta)\mathbf{U}_i^n \mathbb{1}_{x<0} + \mathbf{T}(\theta)\mathbf{U}_j^n \mathbb{1}_{x>0}. \end{cases}$$

Define $\mathbf{w}(t,y) = \widehat{w}(t,x)$ with $y = (\mathbf{V}_j^n\cdot\mathbf{n}_{ij}^n)t + x$. It follows that $\mathbf{w}(y,t)$ satisfies the equation

$$\begin{cases} \partial_t \mathbf{w} + \partial_y \mathfrak{F}(\mathbf{w}) = 0, \\ \mathbf{w}(0,y) = \mathbf{T}(\theta)\mathbf{U}_i^n \mathbb{1}_{y<0} + \mathbf{T}(\theta)\mathbf{U}_j^n \mathbb{1}_{y>0}. \end{cases}$$

Therefore, we conclude that $\mathbf{w}$ satisfies the equation (4.104) and

$$\mathbf{v}(t,x) = \mathbf{T}(\theta)^{-1}\widehat{w}(t,x) = \mathbf{T}(\theta)^{-1}\widehat{w}(t,x) = \mathbf{T}(\theta)^{-1}\mathbf{w}(t,(\mathbf{V}_j^n\cdot\mathbf{n}_{ij}^n)t + x).$$

□

Corollary 4.6.26. *The following relation holds*

$$\lambda_{\max}(\mathbf{g}_j^n, \mathbf{n}_{ij}^n, \mathbf{U}_i^n, \mathbf{U}_j^n) = \max\{\lambda_l - (\mathbf{V}_j^n\cdot\mathbf{n}_{ij}^n), \lambda_r - (\mathbf{V}_j^n\cdot\mathbf{n}_{ij}^n)\} \tag{4.105}$$

where λ_l and λ_r is the minimum and maximum "wave" speed of the Riemann problem (4.104).

Note that $\mathfrak{F}(\mathbf{u}) \in \mathbb{R}^{4\times 4}$ and there are 4 eigenvalues for each $\mathbf{u}$: $\lambda_1 = u - a, \lambda_2 = \lambda_3 = u, \lambda_4 = u + a$ with $a = \sqrt{\frac{\gamma p}{\rho}}$. The multiplicity of eigenvalue u is 2 corresponding to contact wave and shear wave. The v does not change across two non-degenerated waves as of result of studying the Riemann invariants relating to 1 and 4 field, or by applying the Rankine-Hugoniot condition for shock waves (see e.g., [74, p. 111]). The primitive variable $(\rho, u, v, p)^\mathsf{T}$ can be used to simplify the computation since the wave type of characteristic fields, genuinely nonlinear or linearly degenerate, and the Riemann invariants of characteristic fields are independent of the choice of variables (see e.g. [30, p. 43, 56]). Therefore finding the solution of Riemann problem (4.104) is exactly the same as finding the solution of 1D Riemann problem

$$\begin{cases} \partial_t \mathbf{w} + \partial_x \mathcal{F}(\mathbf{w}) = 0, \\ \mathbf{w}(0, x) = \mathbf{w}_L \mathbb{1}_{x<0} + \mathbf{w}_R \mathbb{1}_{x>0}, \end{cases} \tag{4.106}$$

where $\mathbf{w} = (\rho, u, E)^\mathsf{T}$, $\mathcal{F} = (\rho u, \rho u^2 + p, u(E + p))^\mathsf{T}$,

$$\mathbf{w}_L = (U^n_{i,1}, \cos\theta U^n_{i,2} + \sin\theta U^n_{i,3}, U^n_{i,4})^\mathsf{T},$$

$$\mathbf{w}_R = (U^n_{j,1}, \cos\theta U^n_{j,2} + \sin\theta U^n_{j,3}, U^n_{j,4})^\mathsf{T}$$

and θ is determined from $\mathbf{n}^n_{ij}$ since $\mathbf{U}^n_j = (U^n_{j,1}, U^n_{j,2}, U^n_{j,3}, U^n_{j,4})^\mathsf{T}$.

Since the wave structure of two non-degenerated waves in (4.104) and (4.106) are the same we can get λ_l and λ_r by solving (4.106) in order to get $\lambda_{\max}$ (see Corollary 4.6.26) which is used to define d_{ij}.

One way is first to find p^* in the star region by applying [74, Theorem 4.1] and then to compare p^* and p_L which is computed from $\mathbf{w}_l$. If $p^* < p_L$, the left wave is a rarefaction wave and hence $\lambda_l = u_L - a_L$; if $p^* \geq p_L$, the left wave is a shock wave and hence $\lambda_l = S_L$ computed as in (4.52) of [74]. Likewise, by comparing p^* and p_r, we can get λ_r.

Numerically the usual way to find p^* is to use Newton-Raphson iterative procedure to

solve $f(p) = 0$. This is because $f(p)$ is monotone increasing and concave down, see (4.37) and (4.38) in [74]. Note that from the formula of $f(p)$ we see that if $u_R - u_L$ is large enough (for fixed p_L and p_R), the vacuum state maybe appear. In all of numerical tests, we do not consider such cases.

A new way to get p^*, which is faster in general, is sated in [42]. The idea is to construct a sequence of shrinking intervals $[p_l^k, p_u^k]$ to approximate p^*. Two end points p_l^{k+1} and p_u^{k+1} of the new interval is chosen as the roots of two quadratic polynomials and given in (4.4) in [42]. The key property is that $p^* \in [p_l^{k+1}, p_u^{k+1}]$. In [42] they also propose a way to get initial upper bound p_u^0 for the case $f(\max\{p_L, p_R\}) < 0$ in (4.3).

4.6.9.2 Nonlinear stability

Lemma 4.6.27. *The set*

$$D := \{\mathbf{u} = (\rho, \rho u, \rho v, E)^\mathsf{T} \in \mathbb{R}^4 : \quad \rho \geq 0, \quad e := E - \rho(u^2 + v^2)/2 \geq 0\} \tag{4.107}$$

is an invariant domain of the Euler equations. Moreover, D is a convex cone in $\mathbb{R}^4$.

Proof. Considering Definition 4.1.1, we only need to show D is a convex set, since by Lemma 4.6.25 and the mean value theorem for integrals, the average of the entropy solution is equal to a state which is in D.

Denote $\mathbf{U}_i = (\rho_i, \rho_i u_i, \rho_i v_i, E_i)^\mathsf{T}, i = 1, 2, 3$. Choose $\alpha \geq 0$ and $\beta \geq 0$. Assume

$\mathbf{U}_1, \mathbf{U}_2 \in D$ and $\mathbf{U}_3 = \alpha\mathbf{U}_1 + \beta\mathbf{U}_2$. Since $E_i = \rho_i e_i + \frac{1}{2}\rho_i[u_i^2 + v_i^2]$, it follows that

$$\begin{aligned}
\rho_3 e_3 =&\rho_3 E_3 - \frac{1}{2}[(\rho_3 u_3)^2 + (\rho_3 v_3)^2] \\
=&(\alpha\rho_1 + \beta\rho_2)(\alpha E_1 + \beta E_2) - \frac{1}{2}[(\alpha\rho_1 u_1 + \beta\rho_2 u_2)^2 + (\alpha\rho_1 v_1 + \beta\rho_2 v_2)^2] \\
=&\alpha^2\{\rho_1 E_1 - \frac{1}{2}[(\rho_1 u_1)^2 + (\rho_1 v_1)^2]\} + \beta^2\{\rho_2 E_2 - \frac{1}{2}[(\rho_2 u_2)^2 + (\rho_2 v_2)^2]\} \\
&+ \alpha\beta[\rho_1 E_2 + \rho_2 E_1 - \rho_1 u_1 \rho_2 u_2 - \rho_1 v_1 \rho_2 v_2] \\
=&\alpha^2\rho_1^2 e_1 + \beta^2\rho_2^2 e_2 + \alpha\beta\rho_1\rho_2[e_2 + \frac{1}{2}(u_2^2 + v_2^2) + e_1 + \frac{1}{2}(u_1^2 + v_1^2) - u_1 u_2 - v_1 v_2] \\
=&\alpha^2\rho_1^2 e_1 + \beta^2\rho_2^2 e_2 + \alpha\beta\rho_1\rho_2[e_1 + e_2 + \frac{1}{2}(u_1 - u_2)^2 + \frac{1}{2}(v_1 - v_2)^2] \\
=&[\alpha\rho_1^2 e_1 + \beta\rho_2^2 e_2][\alpha\rho_1 + \beta\rho_2] + \alpha\beta\rho_1\rho_2[\frac{1}{2}(u_1 - u_2)^2 + \frac{1}{2}(v_1 - v_2)^2]
\end{aligned}$$

Since $\rho_3 = \alpha_1\rho_1 + \beta_2\rho_2 \geq 0$, $e_1 \geq 0$ and $e_2 \geq 0$, we conclude that $e_3 \geq 0$ and hence $\mathbf{U}_3 \in D$. Since $\alpha \geq 0$ and $\beta \geq 0$ are arbitrary, D is a convex cone.

□

Using the scheme (4.50) or the scheme (4.89) to solve Euler equations (4.99), by Theorem 4.6.9 for the invariant set D, we obtain that the density and the internal energy are always positive as stated in the following Corollary.

Corollary 4.6.28. *If Δt^n satisfies the CFL condition* (4.72)*(resp.* (4.96)*), then the solution $\mathbf{u}_h^{n+1}$ obtained from the scheme* (4.50)*(resp.* (4.89)*) for the Euler equations* (4.99) *is in D for any n. In particular,*

$$\rho_i^{n+1} \geq 0, \quad e_i^{n+1} \geq 0, \quad \forall i.$$

4.6.9.3 Entropy inequality

Let s be the specific physical entropy. It is known that $-\rho s$ is convex with respect to the conservative variable $(\rho, \rho u, \rho v, E)$ and is a mathematical entropy to the Euler equations with entropy flux $(-\rho us, -\rho vs)$(see [30, p. 104]). Generally, $-\rho\mathsf{F}(s)$ is a mathematical entropy to the Euler equation with entropy flux $(-\rho u\mathsf{F}(s), -\rho v\mathsf{F}(s))$ under the condition

that $-\rho\mathsf{F}(s)$ is a convex function with respect to the conservative variable $(\rho, \rho u, \rho v, E)$. The convexity of $-\rho\mathsf{F}(s)$ is equivalent to the condition that

$$\mathsf{F}'(s) > 0, \quad \mathsf{F}'(s)\frac{1}{\gamma} - \mathsf{F}''(s) > 0, \tag{4.108}$$

for polytropic ideal gases (see, e.g., [41, 72, 47, 46]).

Theorem 4.6.14 implies the following result.

Corollary 4.6.29. *If Δt^n satisfies the CFL condition* (4.72)*, then the solution of the scheme* (4.50) *solving the 2D Euler equations* (4.99) *satisfies the following discrete entropy inequality*

$$\begin{aligned}\frac{m_i^{n+1}\rho_i^{n+1}\mathsf{F}(s_i^{n+1}) - m_i^n\rho_i^n\mathsf{F}(s_i^n)}{\Delta t^n} \\ + \sum_l \omega_l \int_{\Omega_{t^n,l}} \sum_j \nabla\cdot\{\rho_j^n\mathsf{F}(s_j^n)[(u_j^n, v_j^n)^\mathsf{T} - \mathbf{V}_j^n]\psi_j^{n,l}(\mathbf{x})\}\psi_i^{n,l}(\mathbf{x})\,\mathrm{d}\mathbf{x} \\ + B(\sum_j \rho_j^n\mathsf{F}(s_j^n)\psi_j^n, \psi_i^n) \geq 0.\end{aligned} \tag{4.109}$$

Theorem 4.6.21 implies the following result.

Corollary 4.6.30. *If Δt^n satisfies the CFL condition* (4.96)*, then the solution of the scheme* (4.89) *solving the 2D Euler equations* (4.99) *satisfies the following discrete entropy inequality*

$$\begin{aligned}\frac{\mathfrak{m}_i^{n+1}\rho_i^{n+1}\mathsf{F}(s_i^{n+1}) - \mathfrak{m}_i^n\rho_i^n\mathsf{F}(s_i^n)}{\Delta t^n} \\ + \int_{\Omega_{t^n}} \sum_j \nabla\cdot\{\rho_j^n\mathsf{F}(s_j^n)[(u_j^n, v_j^n)^\mathsf{T} - \mathbf{V}_j^n]\psi_j^n(\mathbf{x})\}\psi_i^n(\mathbf{x})\,\mathrm{d}\mathbf{x} \\ + B(\sum_j \rho_j^n\mathsf{F}(s_j^n)\psi_j^n, \psi_i^n) \geq 0.\end{aligned} \tag{4.110}$$

Remark 4.6.31. *Since $\sum_i \mathbf{c}_{ij}^n = 0$ and $\sum_i d_{ij} = 0$, summing over i, the discrete entropy*

inequality (4.109) *implies that*

$$\sum_i m_i^{n+1}\rho_i^{n+1}\mathsf{F}(s_i^{n+1}) \geq \sum_i m_i^n\rho_i^n\mathsf{F}(s_i^n). \tag{4.111}$$

Corresponding to the scheme (4.110)*, we have*

$$\sum_i \mathfrak{m}_i^{n+1}\rho_i^{n+1}\mathsf{F}(s_i^{n+1}) \geq \sum_i \mathfrak{m}_i^n\rho_i^n\mathsf{F}(s_i^n). \tag{4.112}$$

In particular, for $\mathsf{F}(x) = x$*, it follows that*

$$\sum_i m_i^{n+1}\rho_i^{n+1}s_i^{n+1} \geq \sum_i m_i^n\rho_i^n s_i^n. \tag{4.113}$$

and

$$\sum_i \mathfrak{m}_i^{n+1}\rho_i^{n+1}s_i^{n+1} \geq \sum_i \mathfrak{m}_i^n\rho_i^n s_i^n. \tag{4.114}$$

Note that those inequalities are discrete version of the inequalities $\frac{d}{dt}\int_\Omega \eta(\mathbf{u}(\mathbf{x},t))\,\mathrm{d}\mathbf{x} \leq 0$ *which is a result of entropy inequality with mathematical entropy* $\eta(\mathbf{u})$*.*

4.6.9.4 Minimum entropy principle

Denote $s_j^n := s(\mathbf{U}_j^n) = s_0 + C_v\log(e_i^n/\rho_i^{\gamma-1})$. By choosing a special F, we get the following result.

Theorem 4.6.32. *If* Δt^n *satisfies the CFL condition* (4.72)*, then the solution of the scheme* (4.50) *solving 2D Euler equations* (4.99) *satisfies*

$$s_i^{n+1} \geq \min_{j\in\mathcal{I}(i)} s_j^n. \tag{4.115}$$

In particular, it follows the minimum entropy principle

$$\min_i s_i^{n+1} \geq \min_i s_j^n. \tag{4.116}$$

Proof. Choose $\mathsf{F}(s) := \min\{s - s_0, 0\}$, where $s_0 = \min_{j\in\mathcal{I}(S_i^n)} s_j^n$. Since $(-\rho\mathsf{F}(s), -\rho(u,v)^\mathsf{T}\mathsf{F}(s))$ is an entropy pair, by (4.109), it follows that

$$m_i^{n+1}\rho_i^{n+1}\mathsf{F}(s_i^{n+1}) \geq 0,$$

as a result of the fact that $\mathsf{F}(s_j^n) = 0$ for any $j \in \mathcal{I}(S_i^n)$. Therefore, $s_i^{n+1} \geq s_0 = \min_{j\in\mathcal{I}(S_i^n)} s_j^n$. □

Remark 4.6.33. *The inequality* (4.116) *can also be proved directly from the convex combination* (4.75) *by noticing that* $-\rho s$ *is strictly convex with respect to the variable* $\mathbf{u}$ *(see, e.g, [31, p. 100], [66, p. 112]) and the fact that* $\widehat{\mathbf{U}}(\mathbf{U}_i^n, \mathbf{U}_j^n)$*, as a 1D Lax-Friedrichs approximation, satisfies the cell entropy inequality for all entropy pairs (see, e.g., [72]).*

4.7 Numerical tests

All tests are solved by using $\mathbb{Q}_1$ finite element method with the help of Dealii [1], an open source library designed for numerical computations with finite elements methods.

4.7.1 Given mesh velocity

To test the convergence property of the proposed algorithm, the mesh velocity is chosen to be the characteristic speed which is given explicitly. In this case, the ALE method becomes the usual Lagrangian method, or characteristic method for scalar equation. The problem is stated as follows

$$\begin{cases} \partial_t u(t,\mathbf{x}) + \nabla\cdot(\boldsymbol{\beta} u) = 0, \quad \mathbf{x} = (x,y) \in [0,1]^2 \\ u(0,\mathbf{x}) = x + y, \end{cases} \tag{4.117}$$

where $\boldsymbol{\beta} = (\sin(\pi x)\cos(\pi y)\cos(2\pi t), -\cos(\pi x)\sin(\pi y)\cos(2\pi t))^\mathsf{T}$.

The final time for this test is $T = 0.5$. The reason is that the solution at time $T = 0.5$ is equal to the initial data due to the symmetry of the special chosen velocity field $\boldsymbol{\beta}$, i.e., $u(T,\mathbf{x}) = u(0,\mathbf{x})$.

Since $\nabla\cdot\boldsymbol{\beta} = 0$ and $\nabla\cdot(\boldsymbol{\beta}u) = \boldsymbol{\beta}\cdot\nabla u$, $\boldsymbol{\beta}$ is the characteristic velocity of (4.117). In this test, we choose $\mathbf{V}_h^n = I_h\boldsymbol{\beta}(t^n, \mathbf{x})$. Since $(\mathbf{F}(\mathbf{U}_j^n) - \mathbf{U}_j^n \otimes \mathbf{V}_j^n) = 0$, the forward Euler step in the algorithm (4.89) becomes that

$$\frac{m_i^{n+1}\mathbf{U}_i^{n+1} - m_i^n\mathbf{U}_i^n}{\Delta t^n} - \sum_{j\in\mathcal{I}(S_i^n)} d_{ij}^n\mathbf{U}_j^n = 0.$$

Note that there is no issue with boundary condition since $\boldsymbol{\beta}\cdot\mathbf{n}|_{\partial\Omega_{t^0}} = 0$. Two tests have been done. In the first test, $d_{ij}^n = 0$, i.e., there is no numerical viscosity. It is just Galerkin approximation to test the accuracy in time of the algorithm. The error and convergence are shown in Table 4.1 where the convergences are shown in 2nd and 4th columns. The 3rd order convergence in time is confirmed. Note that there is no space error due to the particular choice of the ALE velocity and the initial data. In the second test, d_{ij}^n is chosen as in (4.24), where $\lambda_{\max}(g_i^n, \mathbf{n}_{ij}^n, \mathbf{U}_i^n, \mathbf{U}_j^n) = |(\boldsymbol{\beta}_i^n - \boldsymbol{\beta}_j^n)\cdot n_{ij}^n|$; hence the viscosity is second-order in space instead of being first-order. The results computed with structured quadrilateral meshes are shown in Table 4.2.

Table 4.1: The convergence of the rotation problem without viscosity with $T = 0.5$ and CFL $= 1.0$

# dofs	L^2		L^1	
	error	convergence	error	convergence
81	7.89709e-04	-	6.46437e-04	-
289	1.41108e-04	2.48	1.15788e-04	2.48
1089	1.72408e-05	3.03	1.41432e-05	3.03
4225	2.14430e-06	3.01	1.75893e-06	3.01
16641	2.75437e-07	2.96	2.25978e-07	2.96
66049	3.44206e-08	3.00	2.82397e-08	3.00

Table 4.2: The convergence of the rotation problem with viscosity with $T = 0.5$ and CFL $= 1.0$

# dofs	L^2		L^1	
	error	convergence	error	convergence
81	1.95461e-02	-	1.30655e-02	-
289	7.35611e-03	1.41	4.28400e-03	1.61
1089	2.60716e-03	1.50	1.22913e-03	1.80
4225	9.14299e-04	1.51	3.28702e-04	1.90
16641	3.21172e-04	1.51	8.50033e-05	1.95
66049	1.13113e-04	1.51	2.16265e-05	1.97

4.7.2 Burgers equation

We consider the following inviscid Burgers equation in two dimensional space

$$\partial_t u + \nabla\cdot\left(\frac{1}{2}u^2\boldsymbol{\beta}\right) = 0, \quad u_0(\mathbf{x}) = \mathbb{1}_{\{\mathbf{x}\in\mathbb{R}^2;\, 0<x_1<1,\, 0<x_2<1\}} \tag{4.118}$$

where $\beta = (1,1)^{\mathsf{T}}$. The solution to this problem at time $t > 0$ and at $\mathbf{x} = (x_1, x_2)$ can be obtained by the rotation of the coordinate and is given as follows. Consider only the case $x_2 \le x_1$ since $u(x_1, x_2, t) = u(x_2, x_1, t)$ for $x_2 > x_1$. Define $\alpha = x_1 - x_2$. Let $\alpha_0 = 1 - \frac{t}{2}$.

(i) If $\alpha > 1$, then $u(x_1, x_2, t) = 0$.

(ii) If $\alpha \le \alpha_0$, then $u(x_1, x_2, t) = \begin{cases} \frac{x_2}{t} & \text{if } 0 \le x_2 < t \\ 1 & \text{if } t \le x_2 < \frac{t}{2} + 1 - \alpha \\ 0 & \text{otherwise,} \end{cases}$

(iii) If $\alpha_0 < \alpha \le 1$, then $u(x_1, x_2, t) = \begin{cases} \frac{x_2}{t} & \text{if } 0 \le x_2 < \sqrt{2t(1-\alpha)} \\ 0 & \text{otherwise.} \end{cases}$

The computation are done up to $T = 1$ in the initial computational domain $\Omega_{t^0} = (-0.25, 1.75)^2$. The boundary of Ω_{t^n} does not move in the time interval $(0, 1)$, i.e., $\partial\Omega_{t^0} = \partial\Omega_{t^n}$ for any $n \ge 0$. The results of the convergence tests are reported in Table 4.3. The

solution u computed on a 128×128 mesh at $T = 1$ are shown in Figure 4.2

Table 4.3: The convergence of the ALE method for the Burgers equation with $T = 1$ and CFL = 0.1

# dofs	L^2		L^1	
	error	convergence	error	convergence
81	5.79E-01	-	6.00E-01	-
289	4.20E-01	0.46	3.88E-01	0.63
1089	2.96E-01	0.51	2.32E-01	0.74
4225	2.14E-01	0.47	1.32E-01	0.82
16641	1.56E-02	0.45	7.40E-02	0.83

4.7.3 KPP problem

This example is a nonlinear scalar conservation law with a non-convex flux

$$\partial_t u + \nabla \cdot \mathbf{f}(u) = 0, \quad u_0(\mathbf{x}) = 3.25\pi \mathbb{1}_{\|\mathbf{x}\|_{\ell^2}<1} + 0.25\pi. \tag{4.119}$$

where $\mathbf{f}(u) = (\sin u, \cos u)^\mathsf{T}$. This test, henceforth referred to as KPP, was proposed in [56]. It is a challenging test for many high-order numerical schemes because the solution has a two-dimensional composite wave structure. The initial computational domain is $\Omega_{t^0} = [-2.5, 1.5] \times [-2.0, 2.5]$. Note that the background velocity is constant and equal to $\beta = \left(\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2}\right)^\mathsf{T}$. It can be shown that the computational domain keeps a rectangular shape in the time interval $(0, 1)$. The computation is done up to time $T = 1$ using $\mathbb{Q}_1$ finite element on structured meshes 128×128 with $CFL = 0.1$. The results are shown in Figure 4.3

By comparing Figure 4.3 and Figure 3.9, one can see that near the left up corner the shock becomes more sharp due to the mesh motion in ALE framework.

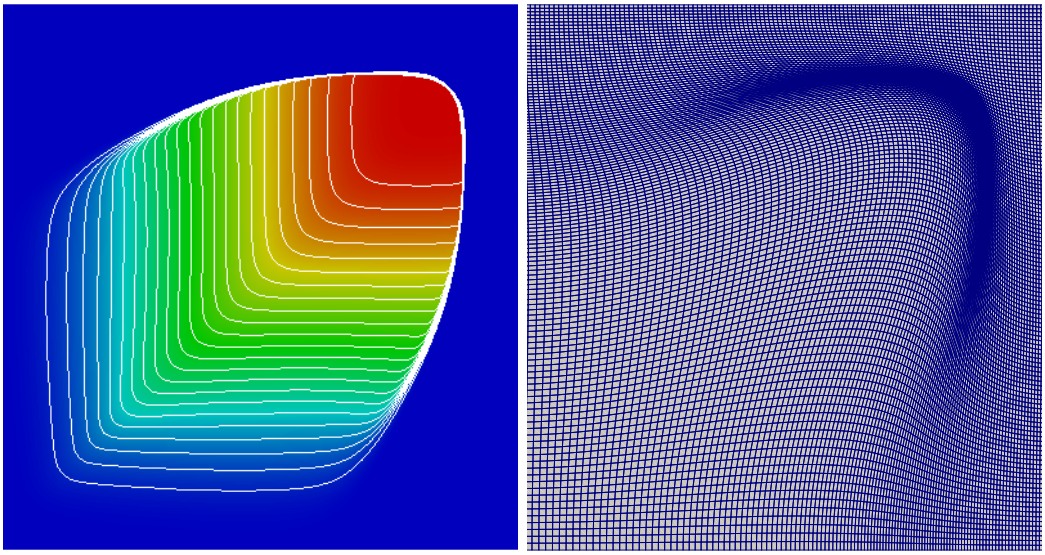


Figure 4.2: The solution and the mesh of the ALE method for the Burgers equation

4.7.4 Sod problem

In the following sections, we solve the compressible Euler equations in $\mathbb{R}^2$, given as in (4.99), with the equation of state, $p = (\gamma - 1)(E - \frac{1}{2}\rho(u^2 + v^2)^2)$ where $\gamma > 1$. The motion of the mesh is done as described in (4.98) with $\mathbf{a}_{\text{Lg}}^{n+1} = \mathbf{a}_i^n + \Delta t(u_h^n, v_h^n)^{\mathsf{T}}(\mathbf{a}_i^n)$ where $\mathbf{U}_i^n = (\rho_i^n, \rho_i^n u_i^n, \rho_i^n v_i^n, E_i^n)$, $u_h^n = \sum_i u_i^n \psi_i^n(\mathbf{x})$ and $v_h^n = \sum_i v_i^n \psi_i^n(\mathbf{x})$.

The first test is the so-called Sod shocktube problem, which is a Riemann problem with the following initial data

$$\rho_0(\mathbf{x}) = 1.0, \quad u_0(\mathbf{x}) = 0.0, \quad p_0(\mathbf{x}) = \mathbb{1}_{x_1<0.5} + 0.1\,\mathbb{1}_{x_1>0.5}. \tag{4.120}$$

and $\gamma = 1.4$ (see e.g. [74, p. 129]). The computational domain at the initial time is the unit square $(0, 1)^2$. Dirichlet boundary conditions are enforced on the left and right sides of the domain and we do not enforce any boundary conditions on the upper and lower sides. The computation is done up to $T = 0.2$. Since no wave reaches the left and the right boundaries in the time interval $0 < t < T$, the computational domain remains a square for

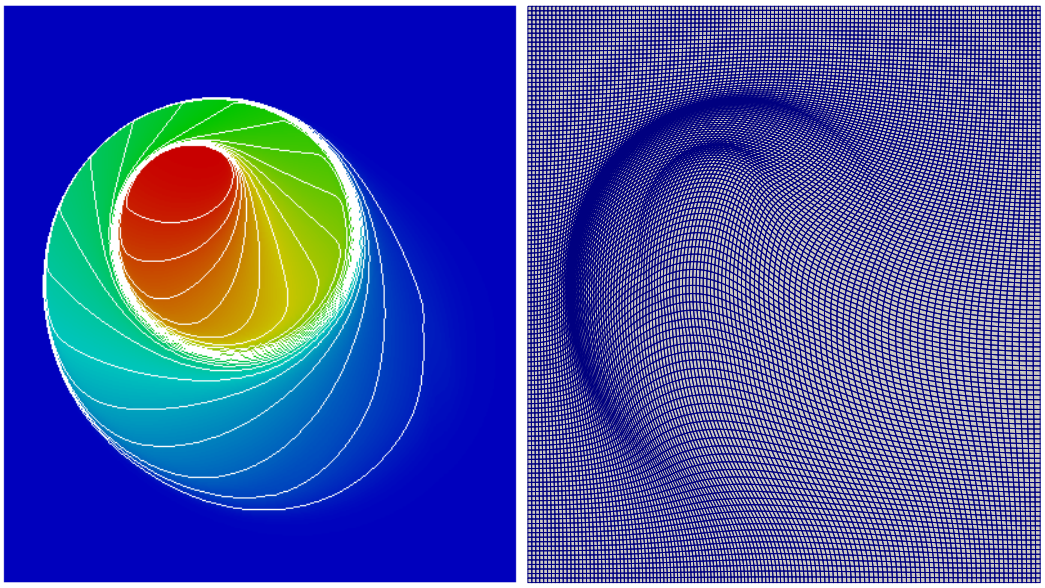


Figure 4.3: The solution and the mesh of the ALE method for the KPP problem

the whole duration of the simulation. The solution being one-dimensional, the convergence tests are done on five meshes with refinements made only along the x_1-direction. Those meshes have 20×4, 40×4, ..., 1280×4 cells. These meshes are uniform at $t = 0$. The results of the convergence test are shown in Table 4.4. We show in this table the L^1- and L^2-norm of the error on the density. The convergence orders are compatible with what is usually obtained in the literature for this problem (≥ 0.62).

Table 4.4: The convergence of the ALE method for the Sod problem with $T = 0.2$ and CFL $= 0.1$

# dofs	L^2		L^1	
	error	convergence	error	convergence
1605	2.47E-02	-	1.51E-02	-
3205	1.84E-02	0.43	9.99E-03	0.60
6405	1.36E-02	0.42	6.42E-03	0.64
12805	1.05E-02	0.39	4.07E-03	0.66

4.7.5 Noh problem

For the Noh problem (see, e.g. [62]) the computational domain at the initial time $t^0 = 0$ is chosen as $\Omega_{t^0} = (-1, 1)^2$ and the initial data is

$$\rho_0(\mathbf{x}) = 1.0, \quad \mathbf{u}_0(\mathbf{x}) = -\frac{\mathbf{x}}{\|\mathbf{x}\|_{\ell^2}} \mathbb{1}_{\|\mathbf{x}\| \neq 0}, \quad p_0(\mathbf{x}) = 10^{-15}. \tag{4.121}$$

A Dirichlet boundary condition is enforced on all the dependent variables on the boundary of the domain for the entire simulation. We use $\gamma = \frac{5}{3}$. The solution to this problem is known (see, e.g. [62]), and the density is given as

$$\rho(t, \mathbf{x}) = 16 \mathbb{1}_{\{\|\mathbf{x}\|_{\ell^2} < \frac{t}{3}\}} + (1 + \frac{t}{\|\mathbf{x}\|_{\ell^2}}) \mathbb{1}_{\{\|\mathbf{x}\|_{\ell^2} > \frac{t}{3}\}}. \tag{4.122}$$

The ALE velocity at the boundary of the computational domain is prescribed to be equal to the fluid velocity, i.e., the boundary moves inwards in the radial direction with speed 1. The final time is chosen to be $T = 0.6$ in order to avoid that the shockwave collides with the moving boundary of the computational domain which happens at $t = \frac{3}{4}$ since the shock moves radially outwards with speed $\frac{1}{3}$.

We show in Table 4.5 the L^1- and the L^2-norm of the error on the density for various meshes which are uniform at $t = 0$: 30×30, 60×60, etc.

Table 4.5: The convergence of the ALE method for the Noh problem with $T = 0.6$ and CFL = 0.2

# dofs	L^2		L^1	
	error	convergence	error	convergence
961	2.59600e+00	-	1.44211e+00	-
3721	1.80963e+00	0.52060	8.44962e-01	0.77122
14641	1.15961e+00	0.64206	4.20578e-01	1.00651
58081	7.66031e-01	0.59816	2.10545e-01	0.99824
231361	5.21009e-01	0.55609	1.06499e-01	0.98329

Preserving the radial symmetry of the solution as best as possible on non-uniform meshes is an important property for Lagrangian hydrocodes in the context of the inertial confinement fusion project, which involves simulating implosion problems, see [11]. In these problems, mesh-induced violation of the spherical symmetry may artificially trigger the Rayleigh-Taylor instability and thereby may hamper the understanding of the real dynamics of the implosion. We show in Figure 4.4 simulations that are done on a uniform mesh composed of 96×96 squares cell for the $\mathbb{Q}_1$ approximation. We compare them with simulations done on a nonuniform mesh constructed as shown in Figure 4.5, where the initial square Ω_{t^0} is divided into four quadrants: in the bottom left quadrant the mesh is composed of 32×32 square cells; in the top left quadrant the mesh is composed of 32×64 rectangular cells; in the top right quadrant the mesh is composed of 64×64 square cells; the bottom right quadrant is composed of 64×32 rectangular cells.

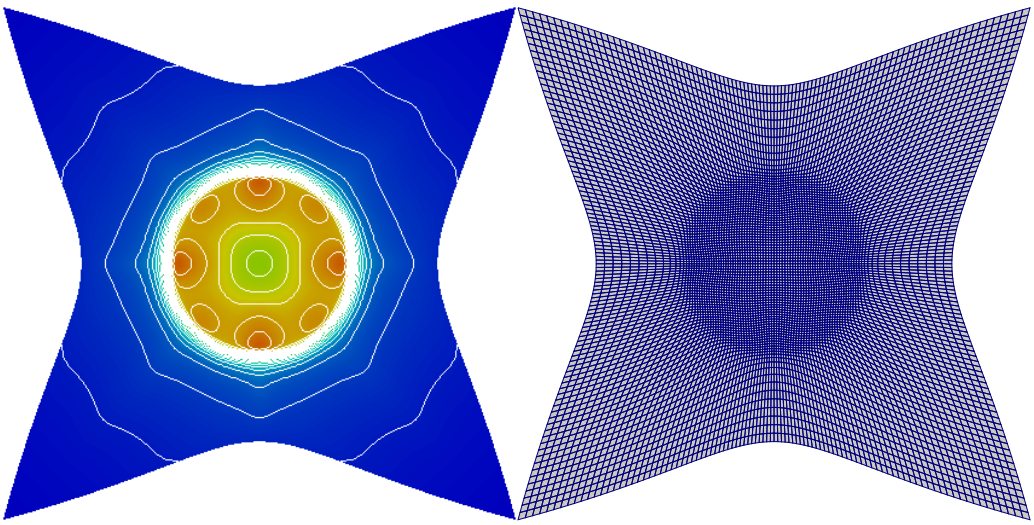


Figure 4.4: The density and the mesh of the Noh problem at $T = 0.6$ on the uniform mesh

We show in the left part of Figure 4.6 a zoom around the center of the computational domain for the $\mathbb{Q}_1$ approximations. We notice a slight motion of the center, but there is no dramatic breakdown of the structure of the solution. The comparison along the line

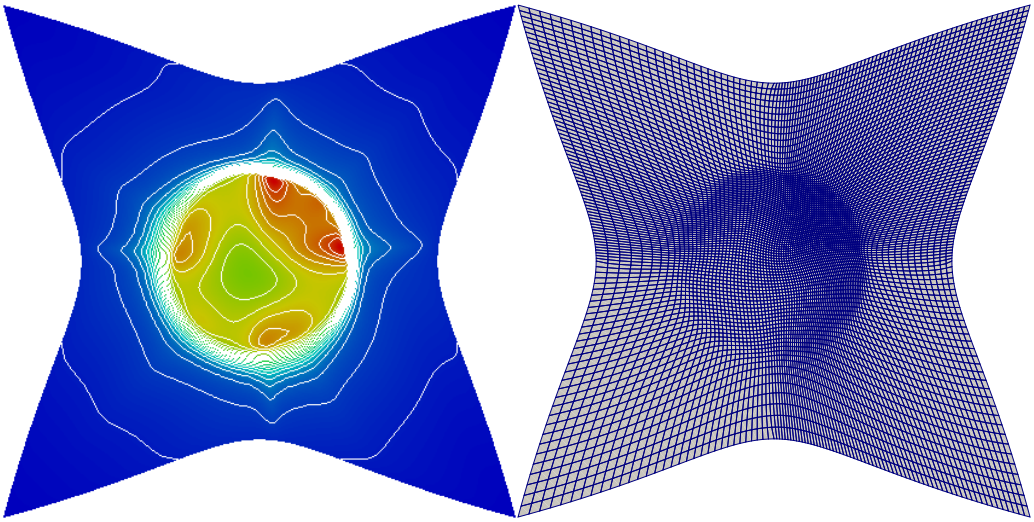


Figure 4.5: The density and the mesh of the Noh problem at $T = 0.6$ on the nonuniform mesh

connecting $(-1,-1)$ and $(1,1)$ is shown in the right part of Figure 4.6. This is a generic test for many Lagrangian hydrocodes, see e.g. [19, §8.4]. We notice a slight break of symmetry, but the solution does not develop any Rayleigh-Taylor-type instability as it is often the case for many other Lagrangian algorithms.

4.7.6 Sedov problem

For the Sedov problem, the initial domain is $\Omega_{t^0} = [-1.2, 1.2]^2$ and the initial data is

$$\rho_0(\mathbf{x}) = 1.0, \quad \mathbf{u}_0(\mathbf{x}) = 0, \quad e_0(\mathbf{x}) = c\phi_{\tilde{\mathbf{a}}}(\mathbf{x}), \tag{4.123}$$

where $t^0 = 0$, c is a constant such that $\int_\Omega e_0 = E_t > 0$, and $\phi_{\tilde{\mathbf{a}}}$ is a Lagrangian basis function corresponding to the node $\tilde{\mathbf{a}}$. In order to place the solution in the center of the domain, assume $(0,0)^\mathsf{T}$ is a vertex of the mesh and $\tilde{\mathbf{a}} = (0,0)^\mathsf{T}$. The Sedov problem is used to simulate the phenomenon of explosion where the pure internal energy is place at the center initially and will be transformed into the kinetic energy along the propagation of a strong shock wave outwardly. The gas is assumed to be polytropic ideal gas with $\gamma = 1.4$. The Dirichlet boundary conditions is added. The final time is $T = 1.0$ which is small

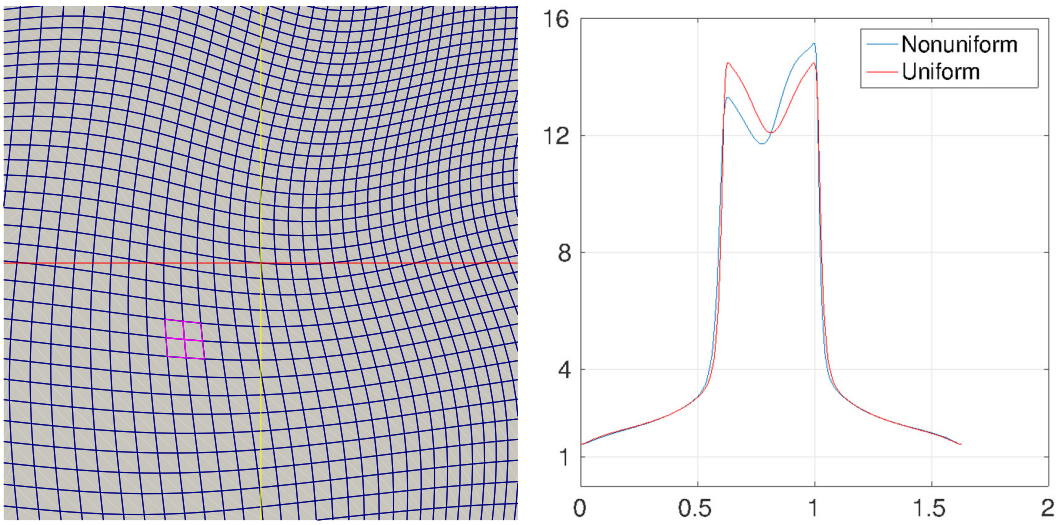


Figure 4.6: The solution of the Noh problem on the nonuniform mesh: Zoom around the center of the mesh (left); cross section along the line connecting $(-1,-1)$ and $(1,1)$ compared with the solution on the uniform mesh (right)

enough so that the velocity of the flow at the boundary is always 0 and the whole domain Ω_{t^0} does not change, i.e., $\Omega_{t^n} = \Omega_{t^0}$. The Lagrangian nodes in the interior of the mesh is moved by the algorithm (4.98) at each time step. The results are shown in Figure 4.7 where the initial mesh consists of 64×64 uniform structured quadrilaterals. The slices of the density passing through $(-1.2, 0)$ and $(1.2, 0)$ are shown in Figure 4.8 on different meshes: 64×64, 128×128, and 256×256.

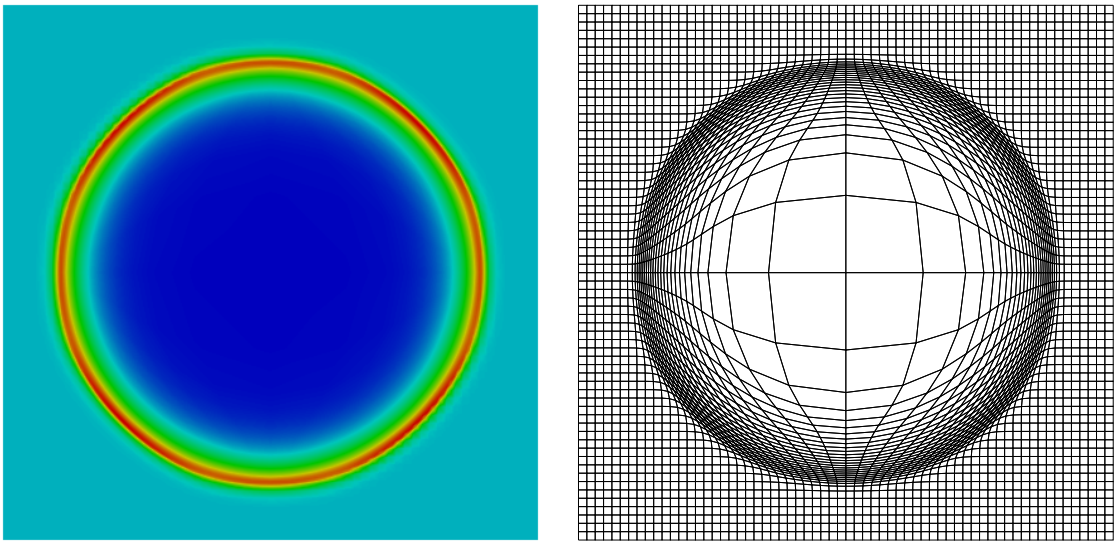


Figure 4.7: The density (left) and the mesh (right) of the Sedov problem with $E_T = 1.0$ at the final time $T = 1.0$

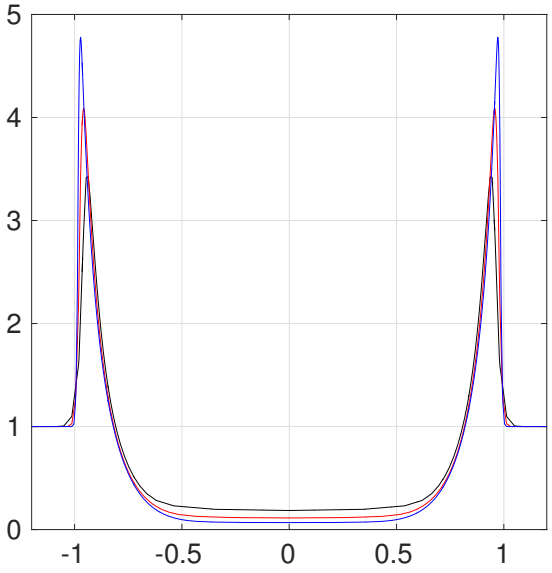


Figure 4.8: The slice of the density of the Sedov problem with $E_T = 1.0$ over the line passing through $(-1.2, 0)$ and $(1.2, 0)$ on the mesh with cells 64×64 (black), 128×128 (red), and 256×256 (blue)

5. CONCLUSION

In this dissertation, we construct and analyze various approximations of nonlinear hyperbolic problems by using continuous finite elements.

In Section 2, the main result is that if the consistent mass matrix is used in the continuous $\mathbb{P}_1$ finite element method for 1D transport equation with numerical viscosity $-\nabla\cdot(\nu h\nabla u)$, for both the Cauchy problem and the periodic boundary value problem, there exist an initial data such that the numerical solution obtained after one forward Euler time step will violate the maximum principle for any $\nu \in D_h$ and $\Delta t > 0$. The results are proved when ν is a constant on each mesh cell. For the Cauchy problem we prove it directly, while for the periodic boundary value problem, it is proved by contradiction. These two results partially answer the question about the necessity of using mass lumping technique for maximum principle preservation.

In Section 3, scalar conservation laws are considered. To obtain a continuous finite element method that keeps the maximum principle and has high-order accuracy at the same time, we use the Zalesak limiter to combine two methods: (i) a low-order method based on the Graph Laplacian and its generalization [36], (ii) a high-order method based on the entropy viscosity, see [40] and [4] for example. A generalization of the original Zalesak limiter is also proposed. The resulting method is both maximum principle satisfying and second order accurate.

In Section 4, systems of conservation laws are considered, where the invariant domain property is studied as a generalization of the maximum principle to systems. In order to combine the advantages of the Eulerian method and that of the Lagrangian method, we choose to work in the ALE framework. One difficulty related to the mesh motion is the requirement of keeping conservation and the invariant domain property at the same time. The requirement of conservation is related to the so-called Discrete Geometric Conservation Laws (DGCL) (see, e.g., [23, 22, 53]) and the invariant domain property is related to

convexity (see, e.g.,[43, 44]). By examining the difference between m_i^{n+1} and m_i^n, we find that we may keep the conservation property by using a quadrature rule in time. For 2D problems, a midpoint rule can be used at least, while for 3D problems, a two points Gaussian quadrature should be used. With the help of an artificial viscosity term proposed in [43] we show that the method keeps the invariant domain property, and other important properties including discrete global conservation, geometric conservation law, discrete entropy inequality. Note that the invariant domain property of Euler equations implies the positivity of density and the internal energy, and the minimum principle of the special entropy. One problem of this algorithm is that it is difficult to extend it to higher order SSPRK methods. It looks like the exact conservation property and invariant domain property are not compatible for higher order time stepping schemes. For that reason, we introduced an new method which has a global discrete conservation property instead of exact conservation (4.90). This new idea also makes it possible to avoid using quadrature rules and get these properties for all SSPRK methods. Several benchmark test cases from the Euler equations, as a typical prototype of hyperbolic system, are studied in Section 4.7. All numerical results confirms the theory described in dissertation.

REFERENCES

[1] Wolfgang Bangerth, Ralf Hartmann, and Guido Kanschat. Deal.ii–a general purpose object oriented finite element library. *ACM Transactions on Mathematical Software*, 33(4), 2007.

[2] David J. Benson. An efficient, accurate, simple ALE method for nonlinear finite element programs. *Computer Methods in Applied Mechanics and Engineering*, 72(3):305–350, 1989.

[3] Oleg Boiarkine, Dmitri Kuzmin, Čanić, Giovanna Guidoboni, and Andro Mikelić. A positivity-preserving ALE finite element scheme for convection-diffusion equations in moving domains. *Journal of Computational Physics*, 230(8):2896–2914, 2011.

[4] Andrea Bonito, Jean-Luc Guermond, and Bojan Popov. Stability analysis of explicit entropy viscosity methods for non-linear scalar conservation equations. *Mathematics of Computation*, 83:1039–1062, 2014.

[5] Jay P. Boris and David L. Book. Flux-corrected transport. I. shasta, a fluid transport algorithm that works. *Journal of Computational Physics*, 135(2):170–186, 1997.

[6] François Bouchut. *Nonlinear stability of finite volume methods for hyperbolic conservation laws and well-balanced schemes for sources*. Frontiers in mathematics. Birkhäuser, Basel, 2004.

[7] Susanne C. Brenner and L. Ridgway Scott. *The mathematical theory of finite element methods*. Texts in applied mathematics. Springer, New York, 2002.

[8] Erik Burman. On nonlinear artificial viscosity, discrete maximum principle and hyperbolic conservation laws. *Bit Numerical Mathematics*, 47:715–733, 2007.

[9] Erik Burman and Alexandre Ern. Discrete maximum principle for Galerkin approximations of the Laplace operator on arbitrary meshes. *Comptes Rendus Mathematique*, 338(8):641–646, 2004.

[10] Erik Burman and Alexandre Ern. Stabilized Galerkin approximation of convection-diffusion-reaction equations: discrete maximum principle and convergence. *Mathematics of Computation*, 74(252):1637–52, 2005.

[11] E. J. Caramana, Mikhail J. Shashkov, and Paul P. Whalen. Formulations of artificial viscosity for multi-dimensional shock wave computations. *Journal of Computational Physics*, 144(1):70–97, 1998.

[12] Mark A. Christon. The influence of the mass matrix on the dispersive nature of the semi-discrete, second-order wave equation. *Computer Methods in Applied Mechanics and Engineering*, 173:147 – 166, 1999.

[13] Mark A. Christon, Mario J. Martinez, and Thomas E. Voth. Generalized fourier analyses of the advection–diffusion equation–part i: one-dimensional domains. *International Journal for Numerical Methods in Fluids*, 45(8):839–887, 2004.

[14] Kai N. Chueh, Charles C. Conley, and Joel A. Smoller. Positively invariant regions for systems of nonlinear diffusion equations. *Indiana University Mathematics Journal*, 26(2):373–392, 1977.

[15] Ramon Codina. Comparison of some finite element methods for solving the diffusion-convection-reaction equation. *Journal of Computer Methods in Applied Mechanics and Engineering*, 156:185–210, 1998.

[16] Constantine M. Dafermos. *hyperbolic conservation laws in continuum physics.* Grundlehren der mathematischen Wissenschaften. Springer, New York, 2010.

[17] Philip J. Davis. *Interpolation and approximation.* Dover Publications, New York, 1975.

[18] Philip J. Davis. *Circulant matrices.* John Wiley & Sons, New York, 1994.

[19] Veselin A. Dobrev, Tzanio V. Kolev, and Robert N. Rieben. High-order curvilinear finite element methods for Lagrangian hydrodynamics. *SIAM Journal on Scientific Computing*, 34(5):B606–B641, 2012.

[20] Jean Donea and Antonio Huerta. *Finite element methods for flow problems.* Finite Element Methods for Flow Problems. John Wiley & Sons, Chichester, 2003.

[21] Alexandre Ern and Jean-Luc Guermond. *Theory and practice of finite elements.* Applied mathematical sciences. Springer, New York, 2004.

[22] Stéphane Étienne, André Garon, and Dominique H. Pelletier. Perspective on the geometric conservation law and finite element methods for ALE simulations of incompressible flow. *Journal of Computational Physics*, 228(7):2313–2333, 2009.

[23] Charbel Farhat, Philippe Geuzaine, and Céline Grandmont. The discrete geometric conservation law and the nonlinear stability of ALE schemes for the solution of flow problems on moving grids. *Journal of Computational Physics*, 174(2):669–694, 2001.

[24] Miloslav Feistauer, Jiřì Felcman, and Ivan Straškraba. *Mathematical and Computational Methods for Compressible Flow.* Numerical mathematics and scientific computation. Oxford University Press, Clarendon, Oxford, 2003.

[25] Hermano Frid. Maps of convex sets and invariant regions for finite-difference systems of conservation laws. *Archive for Rational Mechanics and Analysis*, 160(3):245–269, 2001.

[26] Lucia Gastaldi. *A Priori* error estimates for the Arbitrary Lagrangian Eulerian formulation with finite elements. *Journal of Numerical Mathematics*, 9(2):123–156, 2001.

[27] Jean-Frédéric Gerbeau, Claude Le Bris, and Tony Lelièvre. *Mathematical methods for the Magnetohydrodynamics of liquid metals.* Numerical Mathematics and Scientific Computation. Oxford Science Publications, 2006.

[28] Norman E. Gibbs, Jr. William G. Poole, and Paul K. Stockmeyer. An algorithm for reducing the bandwidth and profile of a sparse matrix. *SIAM Journal on Numerical Analysis*, 13(2):236–250, 1976.

[29] Vivette Girault and Pierre-Arnaud Raviart. *Finite element methods for navier-stokes equations : theory and algorithms.* Springer Series in Computational Mathematics.

Springer, New York, 1986.

[30] Edwige Godlewski and Pierre-Arnaud Raviart. *Numerical approximation of hyperbolic systems of conservation laws.* Applied mathematical sciences. Springer, New York, 1996.

[31] Edwige Godlewski and Pierre-Arnauld Raviart. *Hyperbolic systems of conservation laws.* Mathématiques et applications. Paris, 1991.

[32] Chris Godsil and Gordon F. Royle. *Algebraic graph theory.* Graduate Texts in Mathematics. Springer, New York, 2001.

[33] Sigal Gottlieb, David Ketcheson, and Chi-Wang Shu. *Strong stability preserving Runge-Kutta and multistep time discretizations.* World Scientific, New Jersey, 2011.

[34] Sigal Gottlieb, Chi-Wang Shu, and Eitan Tadmor. Strong stability-preserving high-order time discretization methods. *SIAM Review*, 43:89–112, 2001.

[35] Philip M. Gresho and Robert L. Sani. *Incompressible Flow And The Finite Element Method, Isothermal Laminar Flow.* Incompressible Flow and the Finite Element Method. John Wiley & Sons, Chichester, 2000.

[36] Jean-Luc Guermond and Murtazo Nazarov. A maximum-principle preserving finite element method for scalar conservation equations. *Computer Methods in Applied Mechanics and Engineering*, 272:198–213, 2014.

[37] Jean-Luc Guermond, Murtazo Nazarov, Bojan Popov, and Yong Yang. A second-order maximum principle preserving Lagrange finite element technique for nonlinear scalar conservation equations. *SIAM Journal on Numerical Analysis*, 52(4):2163–2182, 2014.

[38] Jean-Luc Guermond and Richard Pasquetti. Entropy-based nonlinear viscosity for fourier approximations of conservation laws. *Comptes Rendus Mathematique*, 346(13-14):801–806, 2008.

[39] Jean-Luc Guermond and Richard Pasquetti. A correction technique for the dispersive effects of mass lumping for transport problems. *Computer Methods in Applied Mechanics and Engineering*, 253:186–198, 2013.

[40] Jean-Luc Guermond, Richard Pasquetti, and Bojan Popov. Entropy viscosity method for nonlinear conservation laws. *Journal of Computational Physics*, 230(11):4248–4267, 2011.

[41] Jean-Luc Guermond and Bojan Popov. Viscous regularization of the euler equations and entropy principles. *SIAM Journal on Applied Mathematics*, 74(2):284–305, 2014.

[42] Jean-Luc Guermond and Bojan Popov. Fast estimation of the maximum wave speed in the riemann problem for the euler equations. *arXiv: 1511.02756*, 2015.

[43] Jean-Luc Guermond and Bojan Popov. Invariant domains and first-order continuous finite element approximation for hyperbolic systems. *arXiv:1509.07461*, 2015.

[44] Jean-Luc Guermond, Bojan Popov, Laura Saavedra, and Yong Yang. Invariant domains preserving ALE approximation of hyperbolic systems with continuous finite elements. *SIAM Journal on Scientific Computing*, 2016. Submitted.

[45] Jean-Luc Guermond, Bojan Popov, and Yong Yang. The effect of the consistent mass matrix on the maximum-principle for scalar conservation equations. *Journal of Scientific Computing*, 2016. Submitted.

[46] Ami Harten, Peter D. Lax, C. David Levermore, and William J. Morokoff. Convex entropies and hyperbolicity for general euler equations. *SIAM Journal on Numerical Analysis*, 35(6):2117–2127, 1998.

[47] Amiram Harten. On the symmetric form of systems of conservation laws with entropy. *Journal of Computational Physics*, 49(1):151–164, 1983.

[48] David Hoff. A finite difference scheme for a system of two conservation laws with artificial viscosity. *Mathematics of Computation*, 33(148):1171–1193, 1979.

[49] David Hoff. Invariant regions for systems of conservation laws. *Transactions of the American Mathematical Society*, 289(2):591–610, 1985.

[50] Willem H. Hundsdorfer and Jan G. Verwer. *Numerical solution of time-dependent advection-diffusion-reaction equations*. Springer series in computational mathematics. Springer, New York, 2007.

[51] Akhtar S. Khan and Sujian Huang. *Continuum Theory of Plasticity*. A Wiley Interscience Publication. John Wiley & Sons, Chichester, 1995.

[52] Patrick Knupp and Stanly Steinberg. *Fundamentals of grid generation*. The Fundamentals of Grid Generation. CRC press, Boca Raton, 1994.

[53] Bruno Koobus and Charbel Farhat. Second-order time-accurate and geometrically conservative implicit schemes for flow computations on unstructured dynamic meshes. *Computer Methods in Applied Mechanics and Engineering*, 170(1-2):103–129, 1999.

[54] Dietmar Kröner. *Numerical schemes for conservation laws*. Advances in numerical mathematics. John Wiley & Sons, Chichester, 1997.

[55] Andrei G. Kulikovskii, Nikolai V. Pogorelov, and Andrei Yu. Semenov. *Mathematical aspects of numerical solution of hyperbolic systems*. Chapman & Hall/CRC monographs and surveys in pure and applied mathematics. Chapman & Hall/CRC, Boca Raton, 2001.

[56] Alexander Kurganov, Guergana Petrova, and Bojan Popov. Adaptive semidiscrete central-upwind schemes for nonconvex hyperbolic conservation laws. *SIAM Journal on Scientific Computing*, 29(6):2381–2401, 2007.

[57] Alexander Kurganov and Eitan Tadmor. New high-resolution central schemes for nonlinear conservation laws and convection–diffusion equations. *IMA Journal of Numerical Analysis*, 5:161–182, 2000.

[58] Dmitri Kuzmin. On the design of general-purpose flux limiters for finite element schemes. I. Scalar convection. *Journal of Computational Physics*, 219(2):513–531,

2006.

[59] Dmitri Kuzmin, Rainald Löhner, and Stefan Turek. *Flux–Corrected Transport*. Scientific Computation. Springer, New York, 2005.

[60] Randall J. LeVeque. *Numerical methods for conservation laws*. Lectures in mathematics: ETH Zürich. Birkhäuser, Basel, 1992.

[61] Randall J. LeVeque and Jonathan B. Goodman. TVD schemes in one and two space dimensions. *Lecture in applied mathematics*, 2(51), 1985.

[62] Richard Liska and Burton Wendroff. Comparison of several difference schemes on 1d and 2d test problems for the euler equations. *SIAM Journal on Scientific Computing*, 25(3):995–1017, 2003.

[63] Raphaël Loubère, Pierre-Henri Maire, Mikhail Shashkov, Jérôme Breil, and Stéphane Galera. Reale: a reconnection-based arbitrary–lagrangian–eulerian method. *Journal of Computational Physics*, 229(12):4724–4761, 2010.

[64] Len G. Margolin. Introduction to an Arbitrary Lagrangian Eulerian computing method for all flow speeds. *Journal of Computational Physics*, 135(2):198–202, 1997.

[65] Keith William Morton. *Numerical solution of convection-diffusion problems*. Applied Mathematics and Mathematical Computation. Chapman & Hall, London, 1996.

[66] Antonín Novotný and Ivan Straškraba. *Introduction to the mathematical theory of compressible flow*. Oxford lecture series in mathematics and its applications. Oxford University Press, New York, 2004.

[67] James S. Peery and Daniel E. Carroll. Multi-material ALE methods in unstructured grids. *Computer Methods in Applied Mechanics and Engineering*, 187(3-4):591–619, 2000.

[68] Richard H. Pletcher, John C. Tannehill, and Dale Anderson. *Computational fluid mechanics and heat transfer*. Series in Computational and Physical Processes in Mechanics and Thermal Sciences. Taylor & Francis, Boca Raton, 1997.

[69] Joel Smoller. *Shock waves and reaction-diffusion equations*. Die Grundlehren der mathematischen Wissenschaften. Springer, New York, 1983.

[70] Peter K. Sweby. High resolution schemes using flux limiters for hyperbolic conservation laws. *SIAM Journal on Numerical Analysis*, 21(5):995–1011, 1984.

[71] Masahisa Tabata. A theoretical and computational study of upwind-type finite element methods. In *Patterns and Waves Qualitative Analysis of Nonlinear Differential Equations*, volume 18 of *Studies in Mathematics and Its Applications*. Elsevier, 1986.

[72] Eitan Tadmor. A minimum entropy principle in the gas dynamics equations. *Applied Numerical Mathematics*, 2(3):211–219, 1986.

[73] Vidar Thomée. *Galerkin finite element methods for parabolic problems*. Springer, New York, 1997.

[74] Eleuterio F. Toro. *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*. Springer, New York, 2009.

[75] Thomas E. Voth, Mario J. Martinez, and Mark A. Christon. Generalized fourier analyses of the advection–diffusion equation–part ii: two-dimensional domains. *International Journal for Numerical Methods in Fluids*, 45(8):889–920, 2004.

[76] Jacob Waltz, Nathaniel R. Morgan, Thomas R. Canfield, Marc R.J. Charest, L.D. Risinger, and John G. Wohlbier. A three-dimensional finite element Arbitrary Lagrangian–Eulerian method for shock hydrodynamics on unstructured grids. *Computers & Fluids*, 92:172–187, 2014.

[77] B.V. Wells, Michael J. Baines, and Paul Glaister. Generation of arbitrary lagrangian eulerian (ALE) velocities, based on monitor functions, for the solution of compressible fluid equations. *International Journal for Numerical Methods in Fluids*, 47:1375–1381, 2005.

[78] Alan M. Winslow. Numerical solution of the quasilinear poisson equation in a nonuniform triangle mesh. *Journal of computational physics*, 1(2):149–172, 1966.

[79] Zhi Yang and Dimitri Mavriplis. Unstructured dynamic meshes with higher-order time integration schemes for the unsteady Navier-Stokes equations. *American Institute of Aeronautics and Astronautics*, 2005.

[80] Steven T. Zalesak. Fully multidimensional flux-corrected transport algorithms for fluids. *Journal of Computational Physics*, 31(3):335–362, 1979.

[81] Xianyi Zeng and Guglielmo Scovazzi. A frame-invariant vector limiter for flux corrected nodal remap in Arbitrary Lagrangian-Eulerian flow computations. *Journal of Computational Physics*, 270:753–783, 2014.

[82] Olgierd C. Zienkiewicz, Robert L. Taylor, and Perumal Nithiarasu. *The finite element method for fluid dynamics*. Elsevier Butterworth-Heinemann, Oxford, 2005.